ACCELERATED ALGORITHMS FOR COMPOSITE SADDLE-POINT PROBLEMS AND APPLICATIONS

A Dissertation
Presented to
The Academic Faculty

by

Yunlong He

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Computational Science and Engineering

School of Mathematics
Georgia Institute of Technology
December 2014

ACCELERATED ALGORITHMS FOR COMPOSITE SADDLE-POINT PROBLEMS AND APPLICATIONS

Approved by:

Professor Haesun Park,
Committee Chair
School of Computational Science and Engineering
Georgia Institute of Technology

Professor Haomin Zhou
School of Mathematics
Georgia Institute of Technology

Professor Renato D.C. Monteiro,
Advisor
School of Industrial and System Engineering
Georgia Institute of Technology

Professor Sung Ha Kang
School of Mathematics
Georgia Institute of Technology

Professor Le Song
School of Computational Science and Engineering
Georgia Institute of Technology

Date Approved: 7 November 2014

To my parents.

PREFACE

Recent advances in numerical optimization have enabled efficient algorithms for solving complex models motivated by machine learning and image processing applications. This dissertation studies two new algorithms for solving composite saddle-point problems, which are closely related to real-world data analysis applications such as sparse principal component analysis, sparse inverse covariance estimation, truncated collaborative filtering and image recovering. The two algorithms are based on the hybrid proximal extragradient framework and use Nesterov-type accelerated methods to approximately solve the prox subproblems. Both methods achieve optimal iteration-complexity on their associated classes of problems. Experiment results also show that the new methods significantly outperform several state-of-the-art algorithms computationally on many relevant problem instances.

The main theoretical results in this dissertation have been published in two under-review articles [16] and [17].

ACKNOWLEDGEMENTS

My past five years at Georgia Tech have been a wonderful journey – one where I have grown both academically and personally. This unforgettable journey would not have been possible without the support of many people that I have interacted with over the years.

First, I am extremely grateful to have been under the guidance of two great advisors, Dr. Renato D.C. Monteiro and Dr. Haesun Park. They have not only taught me knowledge about numerical algorithms, machine learning and optimization, but also shared without any reservation to me how to do research and write academic papers. Sometimes, projects went into stalemates in my hands and made me frustrated. Luckily, my two advisors were always incredibly patient and kind. Their encouragement made me keep going on the road of academic research. Moreover, their relentless passionate for research and new knowledge has been inspiring me during the past five years and will direct me to pursue lifelong research.

I would like to thank my committee members, Prof. Haomin Zhou, Prof. Sung Ha Kang and Prof. Le Song. Without their comments and suggestions, I would not have finished my work. I would also like thank Prof. Hongyuan Zha and Prof. Jack Poulson. They shared with me a lot of insightful ideas when I worked with them on the XDATA projects. Thanks to my internship at NEC Labs, I met two friends and collaborators, Prof. Yanjun Qi and Dr. Koray Kavukcuoglu who were my mentors there. I appreciate their guidance and discussion on topics of information retrieval, latent factor models, computer vision and deep learning. It is them who brought me into the machine learning community.

I would like to thank the faculty and staff members at School of Mathematics,

especially Prof. Luca Dieci, Prof. John Etnyre, Ms. Klara Grodzinsky, Ms. Cathy Jacobson, Ms. Sharon McDowell and IT staffs for their help on administrations, teaching and many other things that make School of Mathematics a warm home for me. Moreover, I would like to thank my colleagues and friends at Georgia Tech, with whom I shared so manny vivid memories. Their support and accompanying has been extremely important for me. I especially want to thank my friend Ruidong Wang, who is a perfect roommate helped me a lot in many aspects of life during the past five years. I also want to thank my friends from my college and high school, who have offered me great support, courage and happiness over the past a few years.

Finally, I'm deeply grateful to my parents who supported me more than I can imagine. Their altitude for life has always been inspiring me and it is them who taught me to dream and to fight for my dreams. When I have any difficulties in life, they are always there for me and heal me with their relentless love. My gratefulness to them is beyond words.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation considers the composite saddle-point (CSP) problem which is motivated by real-world applications in the areas of machine learning and image processing. Two new accelerated algorithms for solving composite saddle-point problems are introduced.

Due to the two-block structure of the CSP problem, it can be solved by any algorithm belonging to the block-decomposition hybrid proximal extragradient (BD-HPE) framework. The framework consists of a family of inexact proximal point methods for solving a general two-block structured monotone inclusion problem which, at every iteration, solves two prox sub-inclusions according to a certain relative error criterion. By exploiting the fact that the two prox sub-inclusions in the context of the CSP problem are equivalent to two composite convex programs, the first part of this dissertation proposes a new instance of the BD-HPE framework that approximately solves them using an accelerated gradient method. It is shown that the new instance is able to take significantly larger prox stepsizes than other instances from this framework that perform single composite gradient steps for solving the sub-inclusions. As a result, it is shown that the first instance has better iteration-complexity than the latter ones.

The second part of this dissertation introduces a new algorithm for solving a special class of CSP problems. The new algorithm is a special instance of the hybrid proximal extragradient (HPE) framework in which a Nesterov's accelerated variant is used to approximately solve the prox subproblems. One of the advantages of the this method is that it works for any constant choice of proximal stepsize. Moreover, a suitable choice of the latter stepsize yields a method with the best known (accelerated

inner) iteration complexity for the aforementioned class of saddle-point problems. In contrast to the smoothing technique of Nesterov, this new accelerated method does not assume that feasible set is bounded due to its proximal point nature.

Experiment results on both synthetic CSP problems and real-world problems show that the two method significantly outperform several state-of-the-art algorithms.

CHAPTER I

INTRODUCTION

A broad class of optimization, saddle-point (SP), equilibrium and variational inequality problems can be posed as the *monotone inclusion problem*, namely: finding z such that

$$0 \in T(z), \tag{1}$$

where T is a maximal monotone point-to-set operator.

The proximal point method, proposed by Rockafellar [46], is a classical iterative scheme for solving the monotone inclusion problem. The method generates a sequence $\{z_k\}$ according to

$$\|z_k - (\lambda_k T + I)^{-1}(z_{k-1})\| \leq e_k, \qquad \sum_{k=1}^{\infty} e_k < \infty.$$

This method has been used as a generic framework for the design and analysis of several implementable algorithms.

New inexact versions of the proximal point method which uses instead relative error criteria were proposed by Solodov and Svaiter [52, 53, 54, 55]. One of these variants, namely, the hybrid proximal extragradient (HPE) framework studied in [52], was used to develop and analyze block decomposition algorithms (see [36]), and we now briefly discuss this framework. The *exact* proximal point iteration from z with stepsize $\lambda > 0$ is given by $z_+ = (\lambda T + I)^{-1}(z)$, which is equivalent to

$$r \in T(z_+), \qquad \lambda r + z_+ - z = 0. \tag{2}$$

In each step of the HPE, the above *proximal system* is solved inexactly with $(z, \lambda) = (z_{k-1}, \lambda_k)$ to obtain $z_k = z_+$ as follows. For a given constant $\sigma \in [0, 1]$, a triple

$(\tilde{z}, \tilde{r}, \varepsilon) = (\tilde{z}_k, \tilde{r}_k, \varepsilon_k)$ is found such that

$$\tilde{r} \in T^{\varepsilon}(\tilde{z}), \qquad \|\lambda\tilde{r} + \tilde{z} - z\|^2 + 2\lambda\varepsilon \leq \sigma^2\|\tilde{z} - z\|^2, \tag{3}$$

where T^{ε} denotes the ε-enlargement [5] of T (It has the property that $T^{\varepsilon}(z) \supset T(z)$ for each z). Note that this construction relaxes both the inclusion and the equation in (2). Finally, instead of choosing $\tilde{z}$ as the next iterate z_+, the HPE framework computes the next iterate z_+ by means of the *extragradient* step $z_+ = z - \lambda\tilde{r}$. Iteration complexity results for the HPE framework were established in [35] and these results depend on the distance of the initial iterate to the solution set instead of the diameter of the feasible set.

Application of the HPE framework to the iteration-complexity analysis of several zero-order (resp., first-order) methods for solving monotone variational inequalities and monotone inclusions (resp., saddle-point problems) are discussed in [35] and in the subsequent papers [36, 37]. More specifically, by viewing Korpelevich's method as well as Tseng's modified forward-backward splitting (MF-BS) method [58] as special cases of the HPE method, the authors have established in [35, 37] the pointwise and ergodic iteration-complexities of these methods applied to either: monotone variational inequalities, monotone inclusions consisting of the sum of a Lipschitz continuous monotone map and a maximal monotone operator with an easily computable resolvent, and convex-concave saddle-point problems.

A framework of block-decomposition (BD) prox-type algorithms is introduced in [36] for solving the monotone inclusion problem consisting of the sum of a continuous monotone map and a point-to-set maximal monotone operator with a separable two-block structure, namely:

$$0 \in T(x, y) := \begin{pmatrix} F_1(x, y) + A(x) \\ F_2(x, y) + B(y) \end{pmatrix}, \tag{4}$$

and presents a general block-decomposition HPE (BD-HPE) framework in the context of (4), which allows for each one of the single-block proximal subproblems to

be solved in an approximate sense. More specifically, given a pair $((x,y),\lambda) = ((x_{k-1},y_{k-1}),\lambda_k)$, an instance of the BD-HPE framework computes an approximate solution $((\tilde{x},\tilde{y}),(\tilde{r}_x,\tilde{r}_y),\varepsilon)$ of (2) (in the sense of (3)) with T given by (4) by first computing an approximate solution $(\tilde{x},\tilde{r}_x,\varepsilon_x)$ of (2) with $T = F_1(\cdot,y_{k-1}) + A(\cdot)$, then computing an approximate solution $(\tilde{y},\tilde{r}_y,\varepsilon_y)$ of (2) with $T = F_2(\tilde{x},\cdot) + B(\cdot)$, and finally setting $\varepsilon = \varepsilon_x + \varepsilon_y$. Moreover, by showing that any method in this framework is also a special instance of the HPE method, convergence rate results are derived in [36] for the BD-HPE framework based on the ones developed in [35] for the HPE method.

The first part of this dissertation considers the composite saddle-point (CSP) problem

$$\min_{x\in X}\max_{y\in Y} \Psi(x,y) + g_1(x) - g_2(y) \tag{5}$$

where Ψ is a differentiable convex-concave function, g_1 and g_2 are proper closed convex (possibly nonsmooth) functions, $X := \operatorname{dom} g_1$ and $Y := \operatorname{dom} g_2$. Equivalently, the above problem is equivalent to the special case of the inclusion problem (4) in which $(F_1(\cdot,\cdot),F_2(\cdot,\cdot)) = (\nabla_x\Psi(\cdot,\cdot),-\nabla_y\Psi(\cdot,\cdot))$ and $(A,B) = (\partial g_1,\partial g_2)$ for some differentiable convex-concave function Ψ and proper closed convex functions g_1 and g_2. The first main contribution of this dissertation is a new BD-HPE method which exploits the fact that the two prox sub-inclusions are equivalent to composite convex programs. By using a Nesterov-type accelerated method (e.g., [42]) to approximately solve them, the method can choose λ_k (constant and) potentially larger than previous BD-HPE methods. As a result, it is shown that the new method outperforms previous BD-HPE methods both theoretically and computationally in situations where $\max\{L_{xx},L_{yy}\} >> L_{xy}$ where L_{zw} denotes the uniform Lipschitz constant of $\nabla_z\Psi(\cdot,\cdot)$ with respect to w.

The second part of this dissertation considers the special class of composite saddle-point problem

$$\min_{x\in X}\max_{y\in Y}\widehat{\Psi}(x,y) = f(x) + \langle Ax, y\rangle + g_1(x) - g_2(y) \tag{6}$$

where A is a linear operator and f is a differentiable convex function whose gradient is L_f-Lipschitz continuous on X. Since (6) is well-known to be equivalent to monotone inclusion problem (1) with T given by

$$T(x,y) = \partial(\widehat{\Psi}(\cdot,y) - \widehat{\Psi}(x,\cdot))(x,y), \tag{7}$$

any instance of the HPE method, including the ones already discussed above, can be used to solve it. This dissertation presents an accelerated instance of the HPE framework which arbitrarily chooses the stepsize λ and solves (3) with T given by (7) by using a Nesterov's accelerated variant for smooth composite saddle-point problems. Both the outer (i.e., HPE) iteration complexity and the inner (i.e., accelerated variant) iteration complexity are derived for the method in terms of a general stepsize λ. Choosing λ so as to minimize the overall number of inner iterations is the best strategy towards minimizing the overall complexity of the accelerated HPE method. An explicit formula in terms of $\|A\|$, L_f, the distance d_0 of the initial iterate to the set of saddle-points of (6) and the specified tolerances is then derived for such a stepsize. Clearly, since d_0 is not known a priori, the above stepsize can not be computed but an alternative stepsize λ depending only on $\|A\|$ and L_f is provided which is optimal for the most common saddle-point problems of the form (6). Moreover, when the feasible set $X \times Y$ is bounded, the expression for the above optimal stepsize with d_0 replaced by the diameter of $X \times Y$ yields another stepsize which implies (if an appropriate choice of inner product in the (x,y)-space is made) an overall complexity for the accelerated HPE method that is similar to that of Nesterov's smoothing technique (see [40]) for finding an ε-saddle-point of (6). It is worth emphasizing that, in contrast to

Nesterov's smoothing technique of [40], the new accelerated method for solving (6) does not assume that $X \times Y$ is bounded due to its proximal point nature.

1.1 Notation and basic definitions

We denote the sets of real numbers by $\Re$, nonnegative numbers by $\Re_+$ and positive numbers by $\Re_{++}$. For a matrix $W \in \Re^{m\times n}$, we denote its Frobenius norm by $\|W\|_F$, the sum of the absolute values of its entries by $\|W\|_1$ and the sum of its singular values by $\|W\|_*$. Let $\mathcal{S}^n$ denote the space of $n \times n$ real symmetric matrices and S^n_+ denote the cone of symmetric positive matrices. For a matrix $W \in \mathcal{S}^n$, we denote its largest eigenvalue by $\theta_{\max}(W)$. We use $\circ$ to denote the element-wise multiplication between two matrices. For any $z > 0$, define $\log^+(z) := \max(0, \log(z))$. Let $\lceil z \rceil$ denote the smallest integer not less than $z \in \Re$. The n-th unit simplex $\Delta_n \subset \Re^n$ is defined as

$$\Delta_n := \left\{ z \in \Re^n : \sum_{i=1}^{n} z_i = 1, z_i \geq 0, i = 1, \cdots, n \right\}. \tag{8}$$

Throughout this dissertation, we let $\mathcal{Z}$ denote a finite dimensional inner product space with associated inner product denoted by $\langle \cdot, \cdot \rangle$ and the induced norm denoted by $\| \cdot \|$. For a given set $\Omega \subset \mathcal{Z}$, the diameter D_Ω of Ω is defined as

$$D_\Omega := \sup\{\|z - \tilde{z}\| : z, \tilde{z} \in \Omega\} \tag{9}$$

and the indicator function $\mathcal{I}_\Omega : \mathcal{Z} \to (-\infty, \infty]$ of Ω is defined as

$$\mathcal{I}_\Omega(z) := \begin{cases} 0, & z \in \Omega, \\ \infty, & z \notin \Omega. \end{cases}$$

Also, if Ω is nonempty and convex, the *orthogonal projection* $P_\Omega : \mathcal{Z} \to \mathcal{Z}$ onto Ω is defined as

$$P_\Omega(z) := \text{argmin}_{\tilde{z}\in\Omega}\|\tilde{z} - z\| \quad \forall z \in \mathcal{Z}.$$

A relation $T \subseteq \mathcal{Z} \times \mathcal{Z}$ can be identified with a point-to-set operator $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ in which

$$T(z) := \{v \in \mathcal{Z} : (z, v) \in T\}, \quad \forall z \in \mathcal{Z}.$$

Note that the relation T is then the same as the graph of the point-to-set operator T defined as

$$Gr(T) := \{(z, v) \in \mathcal{Z} \times \mathcal{Z} : v \in T(z)\}.$$

An operator $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ is *monotone* if

$$\langle r - \tilde{r}, z - \tilde{z} \rangle \geq 0, \qquad \forall (z, r), (\tilde{z}, \tilde{r}) \in Gr(T).$$

Moreover, T is *maximal monotone* if it is monotone and maximal in the family of monotone operators with respect to the partial order of inclusion, i.e., $S : \mathcal{Z} \rightrightarrows \mathcal{Z}$ monotone and $Gr(S) \supset Gr(T)$ implies that $S = T$. Given a scalar ε, the ε-enlargement of a point-to-set operator $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ is the point-to-set operator $T^{\varepsilon} : \mathcal{Z} \rightrightarrows \mathcal{Z}$ defined as

$$T^{\varepsilon}(z) := \{r \in \mathcal{Z} \mid \langle z - \tilde{z}, r - \tilde{r} \rangle \geq -\varepsilon, \quad \forall \tilde{z} \in \mathcal{Z},\ \forall \tilde{r} \in T(\tilde{z})\}, \quad \forall z \in \mathcal{Z}. \tag{10}$$

The effective domain of a function $f : \mathcal{Z} \to [-\infty, \infty]$ is defined as $\operatorname{dom} f := \{z \in \mathcal{Z} : f(z) < \infty\}$. Moreover, if f is differentiable at point $\tilde{z}$ such that $f(\tilde{z}) \in \Re$, its first-order (affine) approximation at $\tilde{z}$ is defined as

$$l_f(z; \tilde{z}) : = f(\tilde{z}) + \langle \nabla f(\tilde{z}), z - \tilde{z} \rangle \qquad \forall z \in \mathcal{Z}. \tag{11}$$

The conjugate f^* of f is the function $f^* : \mathcal{Z} \to [-\infty, \infty]$ defined as

$$f^*(v) := \sup_{z \in \mathcal{Z}} \langle v, z \rangle - f(z), \quad \forall v \in \mathcal{Z}.$$

Given a scalar $\varepsilon \geq 0$, the ε-subdifferential of a function $f : \mathcal{Z} \to [-\infty, +\infty]$ is the operator $\partial_{\varepsilon} f : \mathcal{Z} \rightrightarrows \mathcal{Z}$ defined as

$$\partial_{\varepsilon} f(z) := \{v \mid f(\tilde{z}) \geq f(z) + \langle \tilde{z} - z, v \rangle - \varepsilon,\ \forall \tilde{z} \in \mathcal{Z}\}, \quad \forall z \in \mathcal{Z}. \tag{12}$$

When $\varepsilon = 0$, the operator $\partial_\varepsilon f$ is simply denoted by ∂f and is referred to as the subdifferential of f. The operator ∂f is trivially monotone if f is proper. If f is a proper closed convex function, then ∂f is maximal monotone [45].

The following result lists some useful properties about the ε-subdifferential of a proper convex function.

Proposition 1.1.1. *Let $f : \mathcal{Z} \to [-\infty, +\infty]$ be a proper convex function. Then*

(a) $\partial_\varepsilon f(z) \subseteq (\partial f)^\varepsilon(z)$ for any $\varepsilon \geq 0$ and $z \in \mathcal{Z}$;

(b) if $v \in \partial f(z)$ and $f(\tilde{z}) < \infty$, then $v \in \partial_\varepsilon f(\tilde{z})$, where $\varepsilon := f(\tilde{z}) - [f(z) + \langle \tilde{z} - z, v \rangle] \geq 0$;

(c) if, in addition, f is closed, then $v \in \partial f(z)$ is equivalent to $z \in \partial f^*(v)$.

The domain of a point-to-point map F is denoted by $\operatorname{Dom} F$. For a constant $L \geq 0$, a map $F : \operatorname{Dom} F \subseteq \mathcal{Z} \to \mathcal{Z}$ is said to be L-Lipschitz continuous on $\Omega \subseteq \operatorname{Dom} F$ if

$$\|F(z) - F(\tilde{z})\| \leq L\|z - \tilde{z}\| \quad \forall z, \tilde{z} \in \Omega; \tag{13}$$

moreover, if in addition $\Omega = \operatorname{Dom} F$, we will simply say that F is L-Lipschitz continuous.

The following result gives a characterization of a strongly convex function in terms of its conjugate.

Proposition 1.1.2. *For a scalar $\beta > 0$ and a proper closed convex function $f : \mathcal{Z} \to [-\infty, \infty]$, the following two properties are equivalent:*

(a) f is strongly convex with modulus β;

(b) f^* is differentiable everywhere and ∇f^* is $1/\beta$-Lipschitz continuous.

Proof. This proposition is equivalent to Proposition 12.60 of [47] in view of the well-known fact that $f = f^{**}$. □

CHAPTER II

COMPOSITE SADDLE-POINT PROBLEMS

2.1 Saddle-point problems

This section presents some basic facts about the saddle-point problem and a notion of an approximate saddle-point.

Let $\mathcal{X}$ and $\mathcal{Y}$ denote finite dimensional inner product spaces with associated inner products both denoted by $\langle \cdot, \cdot \rangle$ and associated norms both denoted by $\|\cdot\|$. We endow the product space $\mathcal{X} \times \mathcal{Y}$ with the canonical inner product defined as

$$\langle (x, y), (\tilde{x}, \tilde{y}) \rangle = \langle x, \tilde{x} \rangle + \langle y, \tilde{y} \rangle, \quad \forall (x, y), (\tilde{x}, \tilde{y}) \in \mathcal{X} \times \mathcal{Y}.. \tag{14}$$

The associated norm, also denoted by $\|\cdot\|$ for shortness, is then given by

$$\|(x, y)\| = \sqrt{\|x\|^2 + \|y\|^2}, \quad \forall (x, y) \in \mathcal{X} \times \mathcal{Y}.$$

We will now review the saddle-point problem and some of its basic properties. Given two nonempty convex sets $X \subseteq \mathcal{X}$ and $Y \subseteq \mathcal{Y}$, we consider throughout this section a function $\widehat{\Psi} : \mathcal{X} \times \mathcal{Y} \to [-\infty, +\infty]$ satisfying the following condition:

A.1) $\widehat{\Psi}(x, y)$ is finite-valued on $X \times Y$ and

$$\widehat{\Psi}(x, y) = \begin{cases} \infty, & x \notin X, \\ -\infty, & x \in X, y \notin Y. \end{cases} \tag{15}$$

The saddle-point problem determined by the triple $(\widehat{\Psi}; X, Y)$, denoted by $SP(\widehat{\Psi}; X, Y)$, consists of finding a pair $(x, y) \in X \times Y$ such that

$$\widehat{\Psi}(x, \tilde{y}) \leq \widehat{\Psi}(x, y) \leq \widehat{\Psi}(\tilde{x}, y), \quad \forall (\tilde{x}, \tilde{y}) \in X \times Y. \tag{16}$$

Clearly, (x, y) is a saddle-point of $SP(\widehat{\Psi}; X, Y)$ if and only if $(x, y) \in X \times Y$ and

$$(0, 0) \in T(x, y) := \partial[\widehat{\Psi}(\cdot, y) - \widehat{\Psi}(x, \cdot)](x, y). \tag{17}$$

Define the primal and dual functions $p : X \to (-\infty, +\infty]$ and $d : Y \to [-\infty, +\infty)$, respectively, as

$$p(\tilde{x}) = \sup_{\tilde{y} \in Y} \widehat{\Psi}(\tilde{x}, \tilde{y}), \qquad d(\tilde{y}) = \inf_{\tilde{x} \in X} \widehat{\Psi}(\tilde{x}, \tilde{y}), \quad \forall (x, y) \in X \times Y, \tag{18}$$

and consider the pair of optimization problems associated with $SP(\widehat{\Psi}; X, Y)$:

$$p_* := \inf_{\tilde{x} \in X} p(\tilde{x}) = \inf_{\tilde{x} \in X} \sup_{\tilde{y} \in Y} \widehat{\Psi}(\tilde{x}, \tilde{y}) \tag{19}$$

and

$$d_* := \sup_{\tilde{y} \in Y} d(\tilde{y}) = \sup_{\tilde{y} \in Y} \inf_{\tilde{x} \in X} \widehat{\Psi}(\tilde{x}, \tilde{y}) \tag{20}$$

Then, the weak duality inequality says that

$$p(\tilde{x}) \geq d(\tilde{y}), \quad \forall (\tilde{x}, \tilde{y}) \in X \times Y. \tag{21}$$

and equivalently

$$\text{gap}(\tilde{x}, \tilde{y}) := p(\tilde{x}) - d(\tilde{y}) \geq 0, \quad \forall (\tilde{x}, \tilde{y}) \in X \times Y. \tag{22}$$

Moreover, it is well-known that (x, y) is a saddle-point if and only if $(x, y) \in X \times Y$ and $\text{gap}(x, y) = 0$. In view of (21), the latter condition is equivalent to $x \in X$ and $y \in Y$ be optimal solutions of (19) and (20), respectively, and the optimal duality gap $p_* - d_*$ be equal to zero.

We now give a definition of an approximate saddle-point.

Definition 2.1.1. *Given $(\rho, \varepsilon) \in \Re_+ \times \Re_+$, $z = (x, y) \in X \times Y$, $r \in \mathcal{X} \times \mathcal{Y}$ and $\tilde{\varepsilon} \in \Re_+$, the triple $(z, r, \tilde{\varepsilon})$ is called a (ρ, ε)-saddle-point of $SP(\widehat{\Psi}; X, Y)$ if $\|r\| \leq \rho$, $\tilde{\varepsilon} \leq \varepsilon$ and*

$$r \in \partial_{\tilde{\varepsilon}}[\widehat{\Psi}(\cdot, y) - \widehat{\Psi}(x, \cdot)](x, y), \tag{23}$$

Moreover, the pair $(z, \tilde{\varepsilon})$ is a called an ε-saddle-point if $(z, 0, \tilde{\varepsilon})$ is a $(0, \varepsilon)$-saddle-point.

We now make some comments about the notion of a (ρ, ε)-saddle-point. First, being weaker than the notion of an ε-saddle-point in the sense that r can be nonzero, it is suitable for analyzing many algorithms for solving saddle-point problems. In particular, it is a natural notion to consider in the context of HPE-type algorithms such as the ones studied in this dissertation since they generate a sequence $\{(z_k, r_k, \tilde{\varepsilon}_k)\}$ such that $(z, r, \tilde{\varepsilon}) = (z_k, r_k, \tilde{\varepsilon}_k)$ satisfies (23) for every k. Second, although it is based on two errors, namely r and $\tilde{\varepsilon}$, instead of just one single scalar error, these errors naturally arise in the sense that r usually expresses the infeasibility error while ε expresses some sort of functional gap.

In this dissertation, we consider saddle-point problems with two additional assumptions:

A.2) $\widehat{\Psi}(\cdot, y)$ and $-\widehat{\Psi}(x, \cdot)$ are proper closed convex functions for every $(x, y) \in X \times Y$;

A.3) the inclusion (17) has a solution, i.e., $T^{-1}(0) \neq \emptyset$.

A function $\widehat{\Psi} : \mathcal{X} \times \mathcal{Y} \to [-\infty, +\infty]$ satisfying conditions A.1 and A.2 for some nonempty convex sets X and Y is called a *closed convex-concave function on* $X \times Y$. It is well-known that its associated map T defined in (17) is maximal monotone (see for example Theorem 6.3.2 in [1]).

2.2 Composite saddle-point problems

This section describes the problems of interest, namely, the composite saddle-point problem and a special class of composite saddle-point problem.

Let $\Psi : \operatorname{dom} \Psi \subseteq \mathcal{X} \times \mathcal{Y} \to \Re$ and two proper closed convex functions $g_1 : \mathcal{X} \to (-\infty, \infty]$ and $g_2 : \mathcal{Y} \to (-\infty, \infty]$ such that $\operatorname{dom} g_1 \times \operatorname{dom} g_2 \subseteq \operatorname{dom} \Psi$ be given. Also, define

$$X := \operatorname{dom} g_1, \quad Y := \operatorname{dom} g_2, \tag{24}$$

and the function $\widehat{\Psi} : \mathcal{X} \times \mathcal{Y} \to (-\infty, \infty]$ as

$$\widehat{\Psi}(x,y) = \begin{cases} \Psi(x,y) + g_1(x) - g_2(y), & (x,y) \in X \times Y, \\ \infty, & x \notin X, \\ -\infty, & x \in X, y \notin Y. \end{cases} \tag{25}$$

The composite saddle-point (CSP) problem determined by the triple $(\Psi; g_1, g_2)$, denoted by $CSP(\Psi; g_1, g_2)$, is the saddle-point problem $SP(\widehat{\Psi}; X, Y)$ where $\widehat{\Psi}$, X and Y are given by (25) and (24), and the following conditions hold:

B.1) Ψ is differentiable on $\Omega_x \times \Omega_y \supset \operatorname{dom} g_1 \times \operatorname{dom} g_2$, where $\Omega_x \subset \mathcal{X}$ and $\Omega_y \subset \mathcal{Y}$ are closed convex sets;

B.2) the function $\Psi(\cdot, y) - \Psi(x, \cdot) : \operatorname{cl}(X \times Y) \to \Re$ is convex for every $(x, y) \in \operatorname{cl}(X \times Y)$;

B.3) there exists $L_{xy} > 0$ such that

$$\|\nabla_x \Psi(x, \tilde{y}) - \nabla_x \Psi(x, y)\| \leq L_{xy} \|\tilde{y} - y\|, \quad \forall x \in \Omega_x, \forall y, \tilde{y} \in \Omega_y.$$

B.4) there exists $L_{xx} \geq 0$ such that

$$\|\nabla_x \Psi(\tilde{x}, y) - \nabla_x \Psi(x, y)\| \leq L_{xx} \|\tilde{x} - x\|, \quad \forall x, \tilde{x} \in \Omega_x, \forall y \in \Omega_y;$$

B.5) there exists $L_{yy} \geq 0$ such that

$$\|\nabla_y \Psi(x, \tilde{y}) - \nabla_y \Psi(x, y)\| \leq L_{yy} \|\tilde{y} - y\|, \quad \forall x \in \Omega_x, \forall y, \tilde{y} \in \Omega_y.$$

According to the discussion of the association between saddle-point problem and min-max problem in Section 2.1, the notations of above CSP problem and problem (5) are interchangeable throughout this dissertation.

The second part of this dissertation considers a special class of composite saddle-point problem (6), namely, problem $SP(\widehat{\Psi}; X, Y)$ where $\widehat{\Psi}$ has the bilinear structure

$$\widehat{\Psi}(x,y) = f(x) + \langle Ax, y \rangle + g_1(x) - g_2(y), \quad \forall (x,y) \in X \times Y \tag{26}$$

$g_1 : \mathcal{X} \to [-\infty, \infty]$ and $g_2 : \mathcal{Y} \to [-\infty, \infty]$ are proper closed convex functions such that $\operatorname{dom} g_1 = X$ and $\operatorname{dom} g_2 = Y$, and the following conditions hold:

C.1) $A : \mathcal{X} \to \mathcal{Y}$ is a linear operator;

C.2) f is convex on a closed convex set $\Omega \supseteq X$;

C.3) f is differentiable on Ω and ∇f is L_f-Lipschitz continuous on Ω.

Throughout this dissertation, it is assumed that g_1 and g_2 are simple functions in the sense that subproblems of the form

$$\min_{x \in X} \frac{1}{2}\|x - \tilde{x}\|^2 + \lambda g_1(x) \quad \text{and} \quad \min_{y \in Y} \frac{1}{2}\|y - \tilde{y}\|^2 + \lambda g_2(y) \tag{27}$$

are easy to solve for any $\tilde{x}$, $\tilde{y}$ and $\lambda > 0$.

We end this section with a note on the equivalence of the composite saddle-point problem (6) and the following composite optimization problem

$$\min_{x \in X} f(x) + g_1(x) + g_2^*(Ax). \tag{28}$$

Problems of in the form of (28) have recently found many applications in image processing and machine learning. In many of these applications, $f(x)$ is a convex data fidelity term, while $g_1(x)$ and $g_2^*(Ax)$ are certain regularizations, e.g., total variation [48], low rank tensor [57, 24], overlapped group lasso [21, 31], and graph regularization [21, 56]. In the next section, we introduce several machine learning and image processing applications that are formulated as problems (5), (6) and (28).

2.3 Related machine learning and image processing applications

In this section, we introduce several machine learning and image processing applications that can be formulated as the CSP problems introduced in Section 2.2 and existing algorithms for solving them.

2.3.1 Sparse principal component analysis

Principal Component Analysis (PCA) is a classical tool for performing data analysis such as dimensionality reduction, data modeling, feature extraction and other learning tasks. It can be widely used in all kinds of data analysis areas like image feature extraction, gene microarray analysis and document analysis. PCA consists of finding a few orthogonal directions in the data space which preserve the most information in the data. This is done by finding directions that would maximize the variance of the projections of the data points along these directions. However, standard PCA generally produces dense directions (i.e., whose entries are mostly nonzero), and hence are too complex to explain the data set. Instead, a standard approach in the learning community is to pursue sparse directions which in some sense approximate the directions produced by standard PCA. Sparse PCA has a few advantages, namely: i) it can be effectively stored and ii) it allows the simpler interpretation of the inherent structure and important information associated with the data set. For these reasons, sparse PCA is a subject which has received a lot of attention from the learning community in the last decade.

Several formulations and algorithms have been proposed to perform sparse PCA. Zou et al.[62] formulate sparse PCA as a LASSO-type optimization problem. Shen and Huang [50] combine simple linear regression and thresholding to solve a regularized SVD problem, which achieves sparse PCA. D'Aspremont et al.'s DSPCA algorithm [10] consists of solving a semidefinite programming relaxation of a certain formulation of sparse PCA whose solution is then post-processed to yield a sparse principal component (PC). Paper [11] by d'Aspremont et al. proposes a greedy algorithm PathSPCA to solve a new semidefinite programming relaxation and provides a sufficient condition for optimality. ESPCA algorithm in Moghaddam et al. [33] obtains good numerical quality by using a combinatorial greedy method, although their method can be slow on large data set. Their method consists of identifying

an active index set (i.e., the indices corresponding to the nonzero entries of the PC) and then using an algorithm such as power-iteration to obtain the final sparse PC. Journée et al.'s GPower method [22] formulates sparse PCA as a nonconcave maximization problem with a penalty term to achieve sparsity, which is then reduced to an equivalent problem of maximizing a convex function over a compact set. The latter problem is then solved by an algorithm which is essentially a generalization of the power-iteration method. Different deflation methods have been studied in [30], which are used to find multiple sparse PCs sequentially. A different multiple sparse PCA approach is proposed in [27] based on a formulation enforcing near orthogonality of the PCs, which is then solved by an augmented Lagrangian approach. The authors in [18] proposed a simple but effective algorithm for finding a single sparse PC. The algorithm consists of two stages. In the first stage, it identifies an active index set with a desired cardinality corresponding to the nonzero entries of the PC. In the second one, it uses the power iteration method to find the best direction with respect to the active index set. The complexity of this algorithm is proportional to the pre-specified cardinality of the solution, but it can be accelerated by adding multiple indices to the active set in every iteration and optimizing it for sparse matrix.

Given a sample covariance matrix $A \in S^n$, sparse PCA problem aims to find a vector $x \in \Re^n$ such that $x^\top Ax/\|x\|^2$ is maximized, while the number of nonzero entry of x is limited. Among different formulations of sparse PCA problem, this dissertation studies the reformulation introduced in [11] which uses a matrix $X \in \mathcal{S}^n$ to denote $xx^\top$ and solves the following problem:

$$\begin{aligned} &\max_X \langle A, X\rangle - \rho\|X\|_0 \\ s.t. \quad &X \in \mathcal{S}^n, tr(X) = 1, X \succeq 0, rank(X) = 1. \end{aligned} \tag{29}$$

The above problem can be relaxed to the following semidefinite programming (SDP)

problem:

$$\max_X \langle A, X\rangle - \rho\|X\|_1 \tag{30}$$
$$s.t. \quad X \in \mathcal{S}^n, tr(X) = 1, X \succeq 0.$$

which has the following equivalent saddle-point problem formulation:

$$\min_X \max_U \langle X, -A + \rho U\rangle \tag{31}$$
$$s.t. \quad X \in \mathcal{S}^n, tr(X) = 1, X \succeq 0,$$
$$U \in \mathcal{S}^n, |U_{ij}| \leq 1.$$

2.3.2 Sparse inverse covariance estimation

One of the classical problems in multivariate statistics is to estimate the covariance matrix or its inverse. We are more and more often faced with the problem of high dimensional covariance matrix estimation where the dimensionality is large when compared with the sample size. The subsection describes the sparse inverse covariance estimation (SICE) problem [14] which aims to deal with high dimensional covariance matrix estimation. It also finds the conditional dependency among the variables of a Gaussian random vector.

Given a sample covariance matrix $A \in S_+^n$, SICE problem is formulated as the following optimization problem

$$\min_{X \succeq 0} -\log\det(X) + \langle A, X\rangle + \|\Lambda \circ X\|_1, \tag{32}$$

where $\Lambda \in \Re_+^{n\times n}$ is a given regularization parameter matrix and $\circ$ denotes element-wise matrix multiplication. Since zeros in the inverse of covariance matrix correspond to conditional independence in the model, sparse inverse covariance estimation can be used to determine a robust estimate of the covariance matrix, and simultaneously discover the sparse structure in the underlying graphical model. In a recent work on learning the dependency structure of latent factors [19], a SICE problem in the form

of (32) has to be solved at every outer iteration to update the dependency structure of the latent factors. Therefore, an efficient algorithm for solving SICE problem is extremely important for this scenario. A few efficient algorithms have been proposed to solve the problem (32), including a method [28] based on Nesterov's smoothing scheme [40], Alternating Linearization Method (ALM [49]) algorithm which is a variant of ADMM [4], and a method based on quadratic approximation [20].

In this dissertation, we consider the following saddle-point reformulation of SICE problem (32)

$$\min_X \max_U \langle A + \Lambda \circ U, X \rangle - \log\det(X) \tag{33}$$
$$s.t. \quad X \succeq 0, U \in \mathcal{S}^n, |U_{ij}| \leq 1.$$

2.3.3 Truncated collaborative filtering

Recommender system is a specific type of information filtering technique that supports users in their decision-making by predicting the "rating" or "preference" that they would give to an item. It is of great importance for the success of e-commerce and online content providers, and gradually gains popularity in various applications such as Amazon item recommendation, Netflix movie recommendation and Yahoo news recommendation. One approach to design a recommendation system that has been seen wide use is collaborative filtering, which are based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users.

Collaborative filtering approach assumes that each a user u and an item i are associated with latent factors represented by $f_u, g_i \in \Re^k$ respectively and the "rating" that user u would give to item i is $r_{ui} = f_u^\top g_i$. In matrix form, this assumption can be written as

$$R \approx F_U^\top G_I,$$

where $R \in \Re^{U \times I}$ is the rating matrix, $F_U = [f_1, \cdots, f_U] \in \Re^{k \times U}$ and $G_I = [g_1, \cdots, g_I] \in$

$\Re^{k\times I}$ are the matrices of latent factors.

Let $S \in \Re^{U\times I}$ be the binary matrix encoding the missing ratings in matrix R where 0 and 1 indicate "missing" and "observed" respectively. The technique of collaborative filtering predicts the missing values in R by solving the optimization problem

$$\min_{F_U,G_I} \|S \circ (R - F_U^\top G_I)\|_F^2 + \lambda\|F_U\|_F^2 + \lambda\|G_I\|_F^2 \tag{34}$$

for latent factor matrices F_U and G_I and then set $r_{u,i} = f_u^\top g_i$ for all (u, i) such that $S_{u,i} = 0$. Note that in problem (34), regularizations on F_U and G_I are introduced to avoid overfitting problem. Even though problem (34) is not convex with respect to F_U and G_I together, it has been shown in [32] that its solution can be found by solving the following convex optimization problem:

$$\min_X \frac{1}{2}\|S \circ (R - X)\|_F^2 + \lambda\|X\|_*. \tag{35}$$

In many applications of collaborative filtering such as movie recommendation, there is a prior range $[l, u]$ for the ratings in matrix R which, for example, is $[1, 5]$ in the case of Netflix movie recommendation. In most runtime systems in the industry, an ad-hoc method to comply this range is to project the predicted ratings into this predefined range $[l, u]$, which is not the best option from the perspective of achieving the least reconstruction error. Inspired by the work in [23], we consider in this dissertation an extension of problem (35):

$$\min_X \frac{1}{2}\|S \circ (R - X)\|_F^2 + \lambda\|X\|_* + \mathcal{I}(l \le X \le u), \tag{36}$$

which incorporates the prior knowledge of rating range in the objective function and makes sure that the reconstruction error for R is minimized at the same time. Note that this optimization problem can also be viewed as an instance of composite optimization problem (28).

2.3.4 Image recovering with sparsity and total-variance regularizations

Imaging processing plays a very important role today in medical diagnosis. In particular, the use of Magnetic Resonance Imaging (MRI) is an extremely important approach for understanding soft tissue changes within the body in a non-invasive manner. Its use of non-ionizing radio frequency emission for image acquisition is considered safe for repeated use in a clinical setting. Many image processing applications including MRI require solving ill-posed inverse problems to recover high quality images from low-dimensional and noisy observations. These challenging problems necessitate the use of regularization through prior knowledge to capture the geometry of natural signals, images, or videos.

Consider the inverse problem proposed in [29] for recovering an image $z_0 \in \Re^{m\times n}$ from noisy and contaminated observation $z = Rz_0 + \omega \in \Re^{m\times n}$, where $R : \Re^{m\times n} \to \Re^{m\times n}$ is the composition of a convolution operator with a discrete Gaussian filter and a masking operator and $\omega \in \Re^{m\times n}$ represents additive white Gaussian noise. Let $\hat{z} = W\hat{x} \in \Re^{m\times n}$ be the recovered image that we are looking for, where is $W : \Re^{m\times n\times l} \to \Re^{m\times n}$ is the wavelet synthesis operator, redundant coefficients $\hat{x} \in \Re^{m\times n\times l}$ and l is the redundancy level of the wavelet frame. The coefficients $\hat{x}$ can be obtained by solving the following instance of composite optimization problem (28):

$$\min_x \|z - RWx\|^2 + \mu\|x\|_1 + \nu\|Wx\|_{TV}, \tag{37}$$

where $\|u\|_{TV} := \sum_{ij}((\nabla_1 u_{ij})^2 + (\nabla_2 u_{ij})^2)^{1/2}$ is a discrete total variation semi-norm. The first term in the above summand is the data fidelity term, and the second and third terms are regularizations enforcing prior knowledge assumed to be satisfied by the original image. Particularly, the first regularization term induces sparsity on the solution of the problem $\hat{x}$ and the second regularization induces sparsity on the gradient of the restored image such that the prior knowledge that MR images of organs are expected to demonstrate piecewise continuous behavior is included in the

objective function. Several algorithms based on splitting framework were proposed in [29, 44] to solve problem (37).

2.4 Previous works on composite optimization and saddle-point problems

Development and analysis of splitting and block-decomposition (BD) methods is by now a well-developed area, although algorithms which allow a relative error tolerance in the solution of the proximal subproblems have been studied in just a few papers. In particular, Ouorou [43] discusses an ε-proximal decomposition using the ε-subdifferential and a relative error criterion on ε. Projection splitting methods for the sum of arbitrary maximal monotone operators using a particular case of the HPE error tolerance for solving the proximal subproblems were presented in [12, 13]. The use of the HPE method for studying BD methods was first presented in [51]. We observe however that none of these works deal with the derivation of iteration-complexity bounds. More recently, Chambolle and Pock [6] have developed and established iteration-complexity bounds for a BD method, which solves the proximal subproblems exactly, in the context of saddle-point problems with a bilinear coupling.

In the context of variational inequalities, we should mention that prior to [35, 37], Nemirovski [39] has established the ergodic iteration-complexity of Korpelevich's method under the assumption that the feasible set of the problem is bounded, and Nesterov [41] has established the ergodic iteration-complexity of a new dual extrapolation algorithm whose termination depends on the guess of a ball centered at the initial iterate. The algorithm was recently extended to solve problem (6) in [15].

In the context of variational inequalities, Nemirovski [39] has established the ergodic iteration complexity of an extension of Korpelevich's method [25], namely, the Mirror-prox algorithm, under the assumption that the feasible set of the problem is bounded.

Nesterov's smoothing scheme [40] solves problem (6) under the assumption that

X and Y are compact convex sets and g_1 is the indicator function of X. It consists of first approximating the objective function of (6) by a convex differentiable function with Lipschitz continuous gradient and then applying an accelerated gradient-type method (see e.g. [40, 2, 60]) to the resulting approximation problem. It is shown that, if the approximation is properly chosen, the above scheme obtains an ε solution of (6) in at most

$$\mathcal{O}\left(\frac{\|A\|}{\varepsilon} D_X D_Y + \sqrt{\frac{L_f}{\varepsilon}} D_X\right)$$

iterations where D_X and D_Y are the diameters of X and Y. The latter bound is also known to be optimal (see for example the discussion in paragraph (1) of Subsection 1.1 of [7]). The method was also discussed in a recent paper [3].

Chambolle and Pock [6] have developed and established the convergence rate for a primal-dual method for solving problem (6) in the context of $f(x)$ being simple and g_1 being the indicator function of the feasible set X. The recent works on primal-dual algorithms [8, 9, 61] can cope with the same problem (6) (in even more generality, since they treats the infinite dimensional case and more terms). In particular this is made explicit in [9] which, by the way, generalizes the Chambolle and Pock algorithm [6] exactly to the setting of (6). A recent paper [7] considers problem (6) with g_1 being the indicator function of the feasible set X and proposed an accelerated primal-dual algorithm that achieved optimal convergence rate for both cases that the feasible set of the problem is bounded or unbounded. A generalized forward-backward splitting algorithm [44] was recently proposed to solve problems relevant to (6).

CHAPTER III

PRELIMINARIES

This chapter contains three sections. The first section reviews a variant of Nesterov's accelerated method for composite convex optimization problem. The second section describes the HPE framework for the monotone inclusion problem. The third section reviews the BD-HPE framework for the two-block structured monotone inclusion problem.

3.1 Accelerated method for composite convex optimization

This section reviews a variant of Nesterov's accelerated first-order method [40, 60] for solving the composite convex optimization problem. Let $\mathcal{X}$ denote a finite dimensional inner product space with associated inner product and norm denoted by $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ and $\| \cdot \|_{\mathcal{X}}$, respectively. Consider the following composite convex optimization problem

$$\inf p(u) := \psi(u) + g(u) \tag{38}$$

where the functions $\psi : \operatorname{dom} \psi \to \Re$ and $g : \mathcal{X} \to [-\infty, \infty]$ satisfy the following conditions:

D.1) g is a proper closed convex function;

D.2) ψ is convex and differentiable on a closed convex set $\Omega \supseteq X := \operatorname{dom} g$;

D.3) the gradient of the function ψ is L-Lipschitz continuous on Ω.

D.4) for some known constant $\mu \geq 0$, the function g is a μ-strongly convex.

Note that we refer to convex functions as 0-strongly convex functions. This terminology has the benefit of allowing us to treat both the convex and strongly convex

case simultaneously. We note that the extra assumption that the strong convexity of p is all in g is not at all restrictive since it can be easily enforced by moving any strong convexity of ψ to the function g (e.g., by subtracting from and/or adding to these functions a suitable positive multiple of the quadratic function $\|\cdot\|_{\mathcal{X}}^2$). We now explicitly state a variant of Nesterov's accelerated method for solving problem (38), which is due to Tseng (see Algorithm 2 in [60]).

[Algorithm 0] A variant of Nesterov's accelerated algorithm of [60]:

0) Let $u_0 \in \mathcal{X}$ be given and set $\Gamma_0 = 0$, $\tilde{u}_0 = w_0 = P_\Omega(u_0)$, $k = 1$;

1) let $\Gamma_k > \Gamma_{k-1}$ be such that

$$\Gamma_k(\Gamma_{k-1}\mu + 1) = L(\Gamma_k - \Gamma_{k-1})^2 \tag{39}$$

and compute $(u_k, w_k, \tilde{u}_k) \in \Omega \times X \times X$ as

$$u_k := \frac{\Gamma_{k-1}}{\Gamma_k}\tilde{u}_{k-1} + \frac{\Gamma_k - \Gamma_{k-1}}{\Gamma_k}w_{k-1}, \tag{40}$$

$$w_k := \operatorname{argmin} \sum_{i=1}^{k} \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma_k} l_\psi(u; u_i) + g(u) + \frac{1}{2\Gamma_k}\|u - u_0\|_{\mathcal{X}}^2, \tag{41}$$

$$\tilde{u}_k := \frac{\Gamma_{k-1}}{\Gamma_k}\tilde{u}_{k-1} + \frac{\Gamma_k - \Gamma_{k-1}}{\Gamma_k}w_k; \tag{42}$$

2) set $k \leftarrow k + 1$ and go to step 1.

end

We now make a few remarks about the relationship between the above method and Algorithm 2 of [60]. First, the latter method computes w_k as in (41) but with the quadratic term $\|u - u_0\|_{\mathcal{X}}^2/2$ replaced by a general strongly convex function $h(u)$. Second, Algorithm 2 of [60] assumes that X is closed, $\Omega = X$ and $u_0 \in X$ so that $u_0 = \tilde{u}_0 = w_0$. On the other hand, Algorithm 0 can start from any point in $\mathcal{X}$ and can handle problems in which X is not necessarily closed. In fact, its ability to start from any point in $\mathcal{X}$ will be exploited later on in Section 4.3 and Section 5.2.

We now state the main technical result from which the convergence rate of the above variant of Nesterov's accelerated algorithm immediately follows.

Proposition 3.1.1. *The sequences* $\{\Gamma_k\}$, $\{u_k\}$, $\{w_k\}$ *and* $\{\tilde{u}_k\}$ *generated by Algorithm 0 satisfy the following inequalities for any* $k \geq 1$*:*

$$\Gamma_k \geq \frac{1}{L} \max\left\{\frac{k^2}{4}, \quad \left(1+\sqrt{\frac{\mu}{4L}}\right)^{2(k-1)}\right\}, \tag{43}$$

and

$$\Gamma_k p(\tilde{u}_k) + \frac{\Gamma_k \mu + 1}{2}\|u - w_k\|^2 \leq \sum_{i=1}^{k} (\Gamma_i - \Gamma_{i-1})[l_\psi(u; u_i) + g(u)] + \frac{1}{2}\|u - u_0\|_{\mathcal{X}}^2, \quad \forall u \in X. \tag{44}$$

As a consequence, the sequence $\{l_{\psi,k}\}$ *of affine functions defined as*

$$l_{\psi,k}(u) := \sum_{i=1}^{k} \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma_k} l_\psi(u; u_i) \quad \forall u \in \mathcal{X} \tag{45}$$

satisfies

$$l_{\psi,k} \leq \psi, \qquad p(\tilde{u}_k) \leq l_{\psi,k}(u) + g(u) + \frac{1}{2\Gamma_k}\|u - u_0\|_{\mathcal{X}}^2 \quad \forall u \in X. \tag{46}$$

Moreover, if the optimal solution set of (38) *is non-empty, then, for any optimal solution* z_* *of* (38)*, we have*

$$p(\tilde{u}_k) - p(z_*) \leq \frac{1}{2\Gamma_k}\|z_* - u_0\|^2. \tag{47}$$

The proof of this convergence result is similar to the proof of Corollary 3(a) of [60]. For the sake of completeness, we provide its proof here. To prove Proposition 3.1.1, we first prove an intermediate result in Lemma 3.1.2.

Lemma 3.1.2. *Define, for* $k \geq 0$*,*

$$\Lambda_k := \min_{u \in \Omega}\left\{\sum_{i=1}^{k} (\Gamma_i - \Gamma_{i-1})[l_\psi(u; u_i) + g(u)] + \frac{1}{2}\|u - u_0\|_{\mathcal{X}}^2\right\}. \tag{48}$$

Then, for every $k \geq 0$*,*

$$\Lambda_{k+1} - \Lambda_k \geq \Gamma_{k+1} p(\tilde{u}_{k+1}) - \Gamma_k p(\tilde{u}_k). \tag{49}$$

Proof. Since $\Gamma_0 = 0$ and $g(u)$ is μ-strongly convex, the function in the minimization problem (48) is strongly convex with modulus $\Gamma_k\mu + 1$. Therefore, we have

$$\begin{aligned}\Lambda_k + \frac{\Gamma_k\mu + 1}{2}\|w_k - w_{k+1}\|_{\mathcal{X}}^2 &\le \sum_{i=1}^{k}(\Gamma_i - \Gamma_{i-1})[l_\psi(w_{k+1}; u_i) + g(w_{k+1})] + \frac{1}{2}\|w_{k+1} - u_0\|_{\mathcal{X}}^2 \\ &= \Lambda_{k+1} - (\Gamma_{k+1} - \Gamma_k)[l_\psi(w_{k+1}; u_{k+1}) + g(w_{k+1})]. \qquad (50)\end{aligned}$$

Now, using the definition of $\tilde{u}_k$ in (42), the definitions (11) and (38) and the convexity of the function $l_\psi(\cdot; u_{k+1}) + g(\cdot)$, we have

$$\begin{aligned}\Gamma_{k+1}[l_\psi(\tilde{u}_{k+1}; u_{k+1}) + g(\tilde{u}_{k+1})] &\le (\Gamma_{k+1} - \Gamma_k)[l_\psi(w_{k+1}; u_{k+1}) + g(w_{k+1})] + \Gamma_k[l_\psi(\tilde{u}_k; u_{k+1}) + g(\tilde{u}_k)] \\ &\le (\Gamma_{k+1} - \Gamma_k)[l_\psi(w_{k+1}; u_{k+1}) + g(w_{k+1})] + \Gamma_k p(\tilde{u}_k). \qquad (51)\end{aligned}$$

Using the relation (39) and the definitions of u_k and $\tilde{u}_k$ in (40) and (42), we have

$$\|\tilde{u}_{k+1} - u_{k+1}\|^2 = \frac{(\Gamma_{k+1} - \Gamma_k)^2}{\Gamma_{k+1}^2}\|w_{k+1} - w_k\|^2 = \frac{\Gamma_k\mu + 1}{\Gamma_{k+1}L}\|w_{k+1} - w_k\|^2.$$

Therefore, the equality above and the inequalities (50) and (51) imply that

$$\Lambda_{k+1} - \Lambda_k \ge \Gamma_{k+1}[l_\psi(\tilde{u}_{k+1}; u_{k+1}) + g(\tilde{u}_{k+1})] + \frac{\Gamma_{k+1}L}{2}\|\tilde{u}_{k+1} - u_{k+1}\|^2 - \Gamma_k p(\tilde{u}_k).$$

Since ψ is L-Lipschitz continuous on Ω, we have

$$l_\psi(\tilde{u}_{k+1}; u_{k+1}) + \frac{L}{2}\|\tilde{u}_{k+1} - u_{k+1}\|^2 \ge \psi(\tilde{u}_{k+1}),$$

which, together with the above inequality and the definition (38), implies (49).

□

Proof of Proposition 3.1.1. It follows from (49) that the sequence $\{\Lambda_k - \Gamma_k p(\tilde{u}_k)\}$ is nondecreasing, which, together with the definition of Λ_k in (48) and the fact $\Gamma_0 = 0$, implies that

$$\Lambda_k - \Gamma_k p(\tilde{u}_k) \ge \Lambda_0 - \Gamma_0 p(\tilde{u}_0) = \min_{u\in\Omega}\frac{1}{2}\|u - u_0\|_{\mathcal{X}}^2 \ge 0.$$

Inequality (44) then follows from the facts that the function in the minimization problem (48) is strongly convex with modulus $\Gamma_k\mu + 1$ and that w_k is its solution.

Moreover, since the relation (39) implies

$$\begin{aligned}\max\{\Gamma_k, \Gamma_k\Gamma_{k-1}\mu\} &\leq \Gamma_k(\Gamma_{k-1}\mu + 1) = L(\Gamma_k - \Gamma_{k-1})^2 \\ &= L(\Gamma_k^{1/2} - \Gamma_{k-1}^{1/2})^2(\Gamma_k^{1/2} + \Gamma_{k-1}^{1/2})^2 \leq 4L\Gamma_k(\Gamma_k^{1/2} - \Gamma_{k-1}^{1/2})^2, \quad (52)\end{aligned}$$

we have

$$\Gamma_k \geq \max\left\{(\Gamma_{k-1}^{1/2} + \frac{1}{\sqrt{4L}})^2, \Gamma_{k-1}\left(1 + \sqrt{\frac{\mu}{4L}}\right)^2\right\}$$

and hence we obtain inequality (43) by induction. The inequalities in (46) and (47) follow immediately from (44) and the definitions (45) and (11). □

3.2 HPE framework for the monotone inclusion problem

Let $T : \mathcal{Z} \rightrightarrows \mathcal{Z}$ be a maximal monotone operator. The monotone inclusion problem for T consists of finding $z \in \mathcal{Z}$ such that

$$0 \in T(z)\,. \quad (53)$$

We also assume throughout this subsection that this problem has a solution, that is, $T^{-1}(0) \neq \emptyset$.

We next review the HPE framework introduced in [52] for solving the above problem and state the iteration complexity results obtained for it in [35].

[HPE] Hybrid Proximal Extragradient Framework:

0) Let $z_0 \in \mathcal{Z}$ and $0 \leq \sigma < 1$ be given and set $k = 1$;

1) choose $\lambda_k > 0$ and find $\tilde{z}_k, \tilde{r}_k \in \mathcal{Z}$, $\sigma_k \in [0, \sigma]$ and $\varepsilon_k \geq 0$ such that

$$\tilde{r}_k \in T^{\varepsilon_k}(\tilde{z}_k), \qquad \|\lambda_k\tilde{r}_k + \tilde{z}_k - z_{k-1}\|^2 + 2\lambda_k\varepsilon_k \leq \sigma_k^2\|\tilde{z}_k - z_{k-1}\|^2\,; \quad (54)$$

2) set $z_k = z_{k-1} - \lambda_k\tilde{r}_k$, set $k \leftarrow k + 1$, and go to step 1.

end

We now make several remarks about the HPE framework. First, the HPE framework does not specify how to choose λ_k and how to find $\tilde{z}_k$, $\tilde{r}_k$ and ε_k as in (54). The particular choice of λ_k and the algorithm used to compute $\tilde{z}_k, \tilde{r}_k$ and ε_k will depend on the particular implementation of the method and the properties of the operator T. Second, if $\tilde{z} := (\lambda_k T + I)^{-1} z_{k-1}$ is the *exact* proximal point iterate, or equivalently

$$\tilde{r} \in T(\tilde{z}), \tag{55}$$

$$\lambda_k \tilde{r} + \tilde{z} - z_{k-1} = 0, \tag{56}$$

for some $\tilde{r} \in \mathcal{Z}$, then $(\tilde{z}_k, \tilde{r}_k) = (\tilde{z}, \tilde{r})$ and $\varepsilon_k = 0$ satisfies (54). Therefore, the error criterion (54) relaxes the inclusion (55) to $\tilde{r} \in T^{\varepsilon}(\tilde{z})$ and relaxes equation (56) by allowing a small error relative to $\|\tilde{z}_k - z_{k-1}\|$.

We define a sequence of ergodic means $\{\tilde{z}_k^a\}$ associated with $\{\tilde{z}_k\}$ as

$$\tilde{z}_k^a := \frac{1}{\Lambda_k} \sum_{i=1}^{k} \lambda_i \tilde{z}_i, \qquad \text{where} \qquad \Lambda_k := \sum_{i=1}^{k} \lambda_i, \tag{57}$$

and define the sequences of ergodic residuals $\{\tilde{r}_k^a\}$ and $\{\varepsilon_k^a\}$ as

$$\tilde{r}_k^a := \frac{1}{\Lambda_k} \sum_{i=1}^{k} \lambda_i \tilde{r}_i, \qquad \varepsilon_k^a := \frac{1}{\Lambda_k} \sum_{i=1}^{k} \lambda_i (\varepsilon_i + \langle \tilde{z}_i - \tilde{z}_k^a, \tilde{r}_i - \tilde{r}_k^a \rangle). \tag{58}$$

The following result describes the pointwise and ergodic convergence rate properties of the HPE framework. Its proof can be found in Theorem 4.4, Lemma 4.5 and Theorem 4.7 of [35].

Theorem 3.2.1. *Let d_0 denote the distance of z_0 to $T^{-1}(0)$. Then, for every $k \in \mathbb{N}$, the following statements hold:*

(a) (pointwise convergence rate) $\tilde{r}_k \in T^{\varepsilon_k}(\tilde{z}_k)$ and there exists an index $i \leq k$ such that

$$\|\tilde{r}_i\| \leq d_0 \sqrt{\frac{1+\sigma}{1-\sigma} \left(\frac{1}{\sum_{j=1}^{k} \lambda_j^2} \right)}, \qquad \varepsilon_i \leq \frac{\sigma^2 d_0^2 \lambda_i}{2(1-\sigma^2) \sum_{j=1}^{k} \lambda_j^2}.$$

(b) (ergodic convergence rate) $\tilde{r}_k^a \in T^{\varepsilon_k^a}(\tilde{z}_k^a)$ *and*

$$\|\tilde{r}_k^a\| \le \frac{2d_0}{\Lambda_k}, \qquad 0 \le \varepsilon_k^a \le \frac{2d_0^2}{\Lambda_k}\left(1 + \frac{\sigma}{\sqrt{(1-\sigma^2)}}\right).$$

3.3 BD-HPE framework for two-block structured monotone inclusion problem

In this section, we review the general BD-HPE framework proposed in [36] for solving the monotone inclusion problem consisting of the sum of a continuous monotone map and a point-to-set maximal monotone operator with a separable two-block structure. The method introduced in Section 4.1 for solving the composite saddle-point problem will be a special instance of the BD-HPE framework.

The problem of interest in this section is the monotone inclusion problem of finding (x, y) such that

$$(0, 0) \in [F + (A \otimes B)](x, y), \tag{59}$$

or equivalently,

$$0 \in F_1(x, y) + A(x), \quad 0 \in F_2(x, y) + B(y), \tag{60}$$

where $F(x, y) = (F_1(x, y), F_2(x, y)) \in \mathcal{X} \times \mathcal{Y}$ and the following conditions are assumed:

E.1) $A : \mathcal{X} \rightrightarrows \mathcal{X}$ and $B : \mathcal{Y} \rightrightarrows \mathcal{Y}$ are maximal monotone operators;

E.2) $F : \mathrm{Dom}\, F \subset \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ is a continuous map such that $\mathrm{Dom}\, F \supset \mathrm{cl}(\mathrm{Dom}\, A) \times \mathcal{Y}$;

E.3) F is monotone on $\mathrm{Dom}\, A \times \mathrm{Dom}\, B$;

E.4) there exists $L_{xy} > 0$ such that

$$\|F_1(x, y') - F_1(x, y)\| \le L_{xy}\|y' - y\|, \quad \forall x \in \mathrm{Dom}\, A, \ \ \forall y, y' \in \mathcal{Y}. \tag{61}$$

We now make a few remarks about the above assumptions. First, it can be easily seen that E.1 implies that the operator $A \otimes B : \mathcal{X} \times \mathcal{Y} \rightrightarrows \mathcal{X} \times \mathcal{Y}$ defined as

$$(A \otimes B)(x, y) = A(x) \times B(y), \quad \forall (x, y) \in \mathcal{X} \times \mathcal{Y},$$

is maximal monotone. Moreover, in view of the proof of Proposition A.1 of [37], it follows that $F+(A\otimes B)$ is maximal monotone. Second, without loss of generality, we have assumed in E.2 that F is defined in $\mathrm{cl}(\mathrm{dom}\, A)\times\mathcal{Y}$ instead of a set of the form $\mathrm{cl}(\mathrm{dom}\, A)\times\Omega$ for some closet convex set $\Omega\supset\mathrm{dom}\, B$ (e.g., $\Omega=\mathrm{cl}(\mathrm{dom}\, B)$). Indeed, if F were defined on the latter set only, then it would be possible to extend it to the whole set $\mathrm{cl}(\mathrm{dom}\, A)\times\mathcal{Y}$ by considering the extension $(x,y)\in\mathcal{X}\times\mathcal{Y}\to F(x,P_\Omega(y))$, which can be easily seen to satisfy E.2-E.4. Note that evaluation of this extension requires computation of a projection onto Ω. Third, assumption E.4 is needed in order to estimate how much an iterate found by the block decomposition scheme below violates the proximal point equation for (59).

We now present the BD-HPE framework of [36] for solving (59) in which the parameter $\tilde{\sigma}_x$ in the BD-HPE framework of [36] is chosen to be equal to σ_x.

[BD-HPE] Block-decomposition HPE framework:

0) Let $(x_0,y_0)\in\mathcal{X}\times\mathcal{Y}$, $\sigma\in(0,1]$ and $\sigma_x,\sigma_y\in[0,\sigma)$ be given and set $k=1$;

1) choose $\lambda_k>0$ such that

$$\theta_{\max}\left(\begin{bmatrix}\sigma_x^2 & \lambda_k\sigma_x L_{xy}\\ \lambda_k\sigma_x L_{xy} & \sigma_y^2+\lambda_k^2L_{xy}^2\end{bmatrix}\right)\le\sigma^2; \tag{62}$$

2) compute a triple $(\tilde{x}_k,\tilde{a}_k,\varepsilon_k^x)\in\mathcal{X}\times\mathcal{X}\times\Re_+$ such that

$$\tilde{a}_k\in A^{\varepsilon_k^x}(\tilde{x}_k),\quad \|\lambda_k(F_1(\tilde{x}_k,y_{k-1})+\tilde{a}_k)+\tilde{x}_k-x_{k-1}\|^2+2\lambda_k\varepsilon_k^x\le\sigma_x^2\|\tilde{x}_k-x_{k-1}\|^2; \tag{63}$$

3) compute a triple $(\tilde{y}_k,\tilde{b}_k,\varepsilon_k^y)\in\mathcal{Y}\times\mathcal{Y}\times\Re_+$ such that

$$\tilde{b}_k\in B^{\varepsilon_k^y}(\tilde{y}_k),\quad \|\lambda_k(F_2(\tilde{x}_k,\tilde{y}_k)+\tilde{b}_k)+\tilde{y}_k-y_{k-1}\|^2+2\lambda_k\varepsilon_k^y\le\sigma_y^2\|\tilde{y}_k-y_{k-1}\|^2; \tag{64}$$

4) let $(\tilde{r}_k^x,\tilde{r}_k^y)=F(\tilde{x}_k,\tilde{y}_k)+(\tilde{a}_k,\tilde{b}_k)$, and set

$$(x_k,y_k)=(x_{k-1},y_{k-1})-\lambda_k(\tilde{r}_k^x,\tilde{r}_k^y), \tag{65}$$

$k \leftarrow k+1$, and go to step 1.

end

We now make a few remarks about the BD-HPE framework. First, it has been shown in Proposition 3.1 of [36] that any instance of the BD-HPE framework is also an instance of the HPE method applied to the monotone inclusion (59). Second, it is easy to see that condition (62) on λ_k is equivalent to

$$\lambda_k \leq \frac{\sqrt{(\sigma^2-\sigma_x^2)(\sigma^2-\sigma_y^2)}}{\sigma L_{xy}}.$$

Hence, in view of the assumption that $\max\{\sigma_x,\sigma_y\} < \sigma$, it follows that is always possible to choose a stepsize $\lambda_k > 0$ satisfying the above inequality. Third, the above framework does not specify how the triples $(\tilde{x}_k, \tilde{a}_k, \varepsilon_k^x)$ and $(\tilde{y}_k, \tilde{b}_k, \varepsilon_k^y)$ are actually computed in steps 2 and 3, respectively. This degree of freedom is what actually enables it to include many specific instances, which depend on the different ways these two triples are computed.

It follows from the definition of $(\tilde{r}_k^x, \tilde{r}_k^y)$ in step 4 of the BD-HPE framework that

$$(\tilde{r}_k^x, \tilde{r}_k^y) \in [F + (A\otimes B)^{\varepsilon_k^x+\varepsilon_k^y}](\tilde{x}_k,\tilde{y}_k) \subset (F + A\otimes B)^{\varepsilon_k^x+\varepsilon_k^y}(\tilde{x}_k,\tilde{y}_k). \tag{66}$$

In view of (59), we can use the sizes of the residuals $(\tilde{r}_k^x, \tilde{r}_k^y)$ and $\varepsilon_k^x + \varepsilon_k^y$ to measure the quality of the iterate $(\tilde{x}_k, \tilde{y}_k)$ as an approximate solution of (59). The next result (also see Theorem 3.3 of [36]) gives convergence rate bounds for the above residuals and their corresponding ergodic version.

Theorem 3.3.1. *Consider the sequences* $\{(\tilde{x}_k,\tilde{y}_k)\}, \{(\tilde{r}_k^x,\tilde{r}_k^y)\}$ *and* $\{(\varepsilon_k^x,\varepsilon_k^y)\}$ *generated by the BD-HPE framework and define the sequences* $\{(\tilde{x}_k^a,\tilde{y}_k^a)\}$, $\{\tilde{r}_k^a\}$ *and* $\{\varepsilon_k^a\}$ *as*

$$(\tilde{x}_k^a,\tilde{y}_k^a) := \frac{1}{\Lambda_k}\sum_{i=1}^k \lambda_i(\tilde{x}_i,\tilde{y}_i), \qquad \tilde{r}_k^a := \frac{1}{\Lambda_k}\sum_{i=1}^k \lambda_i(\tilde{r}_i^x,\tilde{r}_i^y), \tag{67}$$

and

$$\varepsilon_k^a := \frac{1}{\Lambda_k} \sum_{i=1}^{k} \lambda_i \left\{ \langle (\tilde{x}_i - \tilde{x}_k^a, \tilde{y}_i - \tilde{y}_k^a), (\tilde{r}_i^x, \tilde{r}_i^y) \rangle + \varepsilon_i^x + \varepsilon_i^y \right\}, \tag{68}$$

where $\Lambda_k := \sum_{i=1}^{k} \lambda_i$. *Let* d_0 *denote the distance of the initial point* $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ *to the solution set of (59). Then, the following statements hold for every* $k \in \mathbb{N}$*,:*

(a) (pointwise convergence rate) $(\tilde{r}_k^x, \tilde{r}_k^y) \in (F + A \otimes B)^{\varepsilon_k^x + \varepsilon_k^y}(\tilde{x}_k, \tilde{y}_k)$*, and there exists an index* $i \leq k$ *such that*

$$\|(\tilde{r}_i^x, \tilde{r}_i^y)\| \leq d_0 \sqrt{\frac{1+\sigma}{1-\sigma} \left(\frac{1}{\sum_{j=1}^{k} \lambda_j^2} \right)}, \quad \varepsilon_i^x + \varepsilon_i^y \leq \frac{\sigma^2 d_0^2 \lambda_i}{2(1-\sigma^2) \sum_{j=1}^{k} \lambda_j^2};$$

(b) (ergodic convergence rate) $\tilde{r}_k^a \in [F + (A \otimes B)]^{\varepsilon_k^a}(\tilde{x}_k^a, \tilde{y}_k^a)$*, and*

$$\|\tilde{r}_k^a\| \leq \frac{2d_0}{\Lambda_k}, \qquad \varepsilon_k^a \leq \frac{2d_0^2}{\Lambda_k}(1 + \bar{\eta}),$$

where

$$\bar{\eta} := \frac{2\sqrt{2}\sigma}{1 - \max(\sigma_x, \sigma_y)} \left(1 + \frac{1}{(1-\sigma_y)^2} \right)^{1/2}. \tag{69}$$

In the next chapter, we will use the above result to the specific case where F is a saddle-point operator (see (70)), and A and B are subdifferentials ∂g_1 and ∂g_2, respectively. Moreover, we will instead work with a functional optimality measure naturally associated saddle-point problem defined in terms of the ε-subdifferential (see Definition 2.1.1).

CHAPTER IV

ACCELERATED BLOCK-DECOMPOSITION ALGORITHM FOR COMPOSITE SADDLE-POINT PROBLEMS

This chapter contains four sections. The first section describes a specialization of the BD-HPE framework [36] to the CSP context. It also establishes the iteration-complexity for the latter framework to find a (ρ, ε)-saddle-point. Moreover, it considers the generic problem underlying steps 1 and 2 of the latter framework. The second section states a specific instance of the CSP-BD-HPE framework in which the aforementioned generic problem is solved by performing a single composite gradient step. The main goal of the third section is to derive the iteration-complexity of solving the aforementioned generic problem using the variant of the Nesterov's accelerated method introduced in Section 3.1. The fourth section considers a special instance of the CSP-BD-HPE framework for solving the CSP problem $CSP(\Psi; g_1, g_2)$ in which steps 1 and 2 of the CSP-BD-HPE framework are solved with the aid of the accelerated method applied to the generic problem.

4.1 Block-decomposition framework for composite saddle-point problems (CSP-BD-HPE)

This section describes a specialization of the BD-HPE framework [36] to the CSP context, which we refer to as the CSP-BD-HPE framework. It also establishes the iteration-complexity for the latter framework to find a (ρ, ε)-saddle-point. Moreover, it considers the generic problem underlying steps 1 and 2 of the CSP-BD-HPE framework.

Under the assumptions B.1-B.3, it is well-known that $CSP(\Psi; g_1, g_2)$ is equivalent to (59) with the map $F : \Omega_x \times \Omega_y \to \mathcal{X} \times \mathcal{Y}$ and the two maximal monotone operators $A : \mathcal{X} \rightrightarrows \mathcal{X}$ and $B : \mathcal{Y} \rightrightarrows \mathcal{Y}$ given by

$$F(x,y) := (\nabla_x \Psi(x,y), -\nabla_y \Psi(x,y)), \quad \forall (x,y) \in \Omega_x \times \Omega_y, \tag{70}$$

$$A := \partial g_1, \quad B := \partial g_2. \tag{71}$$

Note that the above map F is not defined on $(\mathrm{cl}X) \times \mathcal{Y}$ and hence does not satisfy condition E.2 and E.4 of the previous subsection. However, as mentioned in the paragraph following conditions E.1-E.4, we can extend the domain of F from $\Omega_x \times \Omega_y$ to $\Omega_x \times \mathcal{Y}$ by considering instead the map $(x,y) \in \Omega_x \times \mathcal{Y} \to F(x, P_{\Omega_y}(y))$, which now satisfies E.2-E.4.

As a consequence of the above observation, it is now possible to state a BD-HPE framework for solving $CSP(\Psi; g_1, g_2)$. Below, for the sake of concreteness, we consider the special case of the BD-HPE framework applied to (59) where F, A and B are as discussed above and the sequence of stepsizes $\{\lambda_k\}$ is assumed to be constant.

[CSP-BD-HPE] BD-HPE framework for soving CSP$(\Psi; \mathbf{g_1}, \mathbf{g_2})$:

0) Let $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$, $\sigma \in (0,1]$ and $\sigma_x, \sigma_y \in [0, \sigma)$ be given, choose $\lambda > 0$ such that

$$\lambda \leq \frac{\sqrt{(\sigma^2 - \sigma_x^2)(\sigma^2 - \sigma_y^2)}}{\sigma L_{xy}} \tag{72}$$

and set $k = 1$;

1) compute a triple $(\tilde{x}, \tilde{a}, \varepsilon^x) \in \mathcal{X} \times \mathcal{X} \times \Re_+$ such that

$$\tilde{a} \in \partial_{\varepsilon^x} g_1(\tilde{x}), \quad \|\lambda(\nabla_x \Psi(\tilde{x}, y'_{k-1}) + \tilde{a}) + \tilde{x} - x_{k-1}\|^2 + 2\lambda\varepsilon^x \leq \sigma_x^2 \|\tilde{x} - x_{k-1}\|^2, \tag{73}$$

where $y'_{k-1} = P_{\Omega_y}(y_{k-1})$, and set $(\tilde{x}_k, \tilde{a}_k, \varepsilon_k^x) = (\tilde{x}, \tilde{a}, \varepsilon^x)$;

2) compute a triple $(\tilde{y}, \tilde{b}, \varepsilon^y) \in \mathcal{Y} \times \mathcal{Y} \times \Re_+$ such that

$$\tilde{b} \in \partial_{\varepsilon^y} g_2(\tilde{y}), \quad \|\lambda(-\nabla_y \Psi(\tilde{x}_k, \tilde{y}) + \tilde{b}) + \tilde{y} - y_{k-1}\|^2 + 2\lambda\varepsilon^y \leq \sigma_y^2 \|\tilde{y} - y_{k-1}\|^2, \tag{74}$$

and set $(\tilde{y}_k, \tilde{b}_k, \varepsilon_k^y) = (\tilde{y}, \tilde{b}, \varepsilon^y)$;

3) let $(\tilde{r}_k^x, \tilde{r}_k^y) = (\nabla_x \Psi(\tilde{x}_k, \tilde{y}_k) + \tilde{a}_k, -\nabla_y \Psi(\tilde{x}_k, \tilde{y}_k) + \tilde{b}_k)$, set

$$(x_k, y_k) = (x_{k-1}, y_{k-1}) - \lambda(\tilde{r}_k^x, \tilde{r}_k^y), \tag{75}$$

$k \leftarrow k+1$, and go to step 1.

end

Specific instances of the CSP-BD-HPE framework will be discussed in Subsection 4.2 and Section 4.4.

Our goal now will be to establish the iteration-complexity of the above framework to obtain a (ρ, ε)-solution of the SP problem (16) (see Definition 2.1.1). We first state the following technical result.

Lemma 4.1.1. *Consider the sequences* $\{(x_k, y_k)\}$, $\{(\tilde{x}_k, \tilde{y}_k)\}$, $\{(\tilde{r}_k^x, \tilde{r}_k^y)\}$ *and* $\{(\varepsilon_k^x, \varepsilon_k^y)\}$ *generated by the CSP-BD-HPE framework and define*

$$\varepsilon_k := \varepsilon_k^x + \varepsilon_k^y. \tag{76}$$

Then the triple $((\tilde{x}_k, \tilde{y}_k), (\tilde{r}_k^x, \tilde{r}_k^y), \varepsilon_k)$ *is a* $(\|(\tilde{r}_k^x, \tilde{r}_k^y)\|, \varepsilon_k)$*-saddle-point of* $CSP(\Psi; g_1, g_2)$*, or equivalently,*

$$(\tilde{r}_k^x, \tilde{r}_k^y) \in \partial_{\varepsilon_k}[\widehat{\Psi}(\cdot, \tilde{y}_k) - \widehat{\Psi}(\tilde{x}_k, \cdot)](\tilde{x}_k, \tilde{y}_k)$$

Proof. In view of step 3) of the CSP-BD-HPE framework, we have

$$\tilde{r}_k^x \in \nabla_x \Psi(\tilde{x}_k, \tilde{y}_k) + \partial_{\varepsilon_k^x} g_1(\tilde{x}_k) \subseteq \partial_{\varepsilon_k^x}[\widehat{\Psi}(\cdot, \tilde{y}_k)](\tilde{x}_k, \tilde{y}_k),$$
$$\tilde{r}_k^y \in -\nabla_y \Psi(\tilde{x}_k, \tilde{y}_k) + \partial_{\varepsilon_k^y} g_2(\tilde{y}_k) \subseteq \partial_{\varepsilon_k^y}[-\widehat{\Psi}(\tilde{x}_k, \cdot)](\tilde{x}_k, \tilde{y}_k),$$

from which we conclude that

$$(\tilde{r}_k^x, \tilde{r}_k^y) \in \left[\partial_{\varepsilon_k^x}[\widehat{\Psi}(\cdot, \tilde{y}_k)](\tilde{x}_k)\right] \times \left[\partial_{\varepsilon_k^y}[-\widehat{\Psi}(\tilde{x}_k, \cdot)](\tilde{y}_k)\right] \subseteq \partial_{\varepsilon_k}[\widehat{\Psi}(\cdot, \tilde{y}_k) - \widehat{\Psi}(\tilde{x}_k, \cdot)](\tilde{x}_k, \tilde{y}_k),$$

where the last inclusion follows from (76) and definition of subgradient. □

The above result shows that the triple $((\tilde{x}_k, \tilde{y}_k), (\tilde{r}_k^x, \tilde{r}_k^y), \varepsilon_k)$ is an approximate solution of $CSP(\Psi; g_1, g_2)$. The result below shows how to construct an approximate solution of $CSP(\Psi; g_1, g_2)$ from the average of the iterates $(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_k, \tilde{y}_k)$.

Lemma 4.1.2. *(Proposition 5.1 of [37]) Let $X \subseteq \mathcal{X}$ and $Y \subseteq \mathcal{Y}$ be given convex sets and let $\Gamma : X \times Y \to \Re$ be a function such that, for each pair $(x, y) \in X \times Y$, the function $\Gamma(\cdot, y) - \Gamma(x, \cdot) : X \times Y \to \Re$ is convex. Suppose that, for $i = 1, \ldots, k$, $(x_i, y_i) \in X \times Y$ and $(r_{x,i}, r_{y,i}) \in \mathcal{X} \times \mathcal{Y}$ satisfies*

$$(r_{x,i}, r_{y,i}) \in \partial_{\varepsilon_i} \Big(\Gamma(\cdot, y_i) - \Gamma(x_i, \cdot) \Big)(x_i, y_i). \tag{77}$$

Let $\alpha_1, \ldots, \alpha_k \geq 0$ be such that $\sum_{i=1}^k \alpha_i = 1$ and define

$$(x^a, y^a) = \sum_{i=1}^k \alpha_i (x_i, y_i), \quad (r_x^a, r_y^a) = \sum_{i=1}^k \alpha_i (r_{x,i}, r_{y,i}), \tag{78}$$

$$\varepsilon^a := \sum_{i=1}^k \alpha_i [\varepsilon_i + \langle x_i - x^a, r_{x,i} \rangle + \langle y_i - y^a, r_{y,i} \rangle] \tag{79}$$

$$\tag{80}$$

Then, $\varepsilon^a \geq 0$ and

$$(r_x^a, r_y^a) \in \partial_{\varepsilon^a} \Big(\Gamma(\cdot, y^a) - \Gamma(x^a, \cdot) \Big)(x^a, y^a). \tag{81}$$

The following iteration-complexity result now follows almost immediately from Lemmas 4.1.1 and 4.1.2 and Theorem 3.3.1.

Theorem 4.1.3. *Consider the sequences $\{(x_k, y_k)\}$, $\{(\tilde{x}_k, \tilde{y}_k)\}$,$\{(\tilde{r}_k^x, \tilde{r}_k^y)\}$ and $\{(\varepsilon_k^x, \varepsilon_k^y)\}$ generated by the BD-HPE framework for $CSP(\Psi; g_1, g_2)$. Let $\{\varepsilon_k\}$ be defined in Lemma 4.1.1 and, for every $k \in \mathbb{N}$, define*

$$(\tilde{x}_k^a, \tilde{y}_k^a) := \frac{1}{k} \sum_{i=1}^k (\tilde{x}_i, \tilde{y}_i), \qquad \tilde{r}_k^a := \frac{1}{k} \sum_{i=1}^k (\tilde{r}_i^x, \tilde{r}_i^y) \tag{82}$$

and

$$\tilde{\varepsilon}_k^a := \frac{1}{k}\sum_{i=1}^{k}[\varepsilon_i + \langle(\tilde{x}_i - \tilde{x}_k^a, \tilde{y}_i - \tilde{y}_k^a), (\tilde{r}_i^x, \tilde{r}_i^y)\rangle]. \tag{83}$$

Then, for every $k \in \mathbb{N}$, the pair $(\tilde{r}_k^a, \tilde{\varepsilon}_k^a)$ is a SP-residual for $(\tilde{x}_k^a, \tilde{y}_k^a)$ with respect to $CSP(\Psi; g_1, g_2)$, or equivalently

$$\tilde{r}_k^a \in \partial_{\tilde{\varepsilon}_k^a}[\widehat{\Psi}(\cdot, \tilde{y}_k^a) - \widehat{\Psi}(\tilde{x}_k^a, \cdot)](\tilde{x}_k^a, \tilde{y}_k^a),$$

and

$$\|\tilde{r}_k^a\| \le \frac{2d_0}{k\lambda}, \qquad \tilde{\varepsilon}_k^a \le \frac{2d_0^2}{k\lambda}(1+\bar{\eta}), \tag{84}$$

where $\bar{\eta}$ is defined in (69) *and d_0 is the distance of (x_0, y_0) to the set of saddle-points of $CSP(\Psi; g_1, g_2)$.*

Proof. It follows from Lemma 4.1.1 and Lemma 4.1.2 with $\Gamma = \hat{\Psi}|_{X\times Y}$, and $(x_i, y_i) = (\tilde{x}_i, \tilde{y}_i)$ and $(r_{x,i}, r_{y,i}) = (\tilde{r}_i^x, \tilde{r}_i^y)$ for $i = 1, \ldots, k$, that

$$\tilde{r}_k^a \in \partial_{\tilde{\varepsilon}_k^a}[\widehat{\Psi}(\cdot, \tilde{y}_k^a) - \widehat{\Psi}(\tilde{x}_k^a, \cdot)](\tilde{x}_k^a, \tilde{y}_k^a).$$

Moreover, (84) follows directly from Theorem 3.3.1 and the fact that the CSP-BD-HPE framework is a special case of the BD-HPE framework in which $\lambda_k = \lambda$ for all k. □

We end this section by describing a generic problem underlying the computation of the triples as in steps 1 and 2 of the CSP-BD-HPE framework. Let $\mathcal{Z}$ be an inner product space and $\tilde{f} : \operatorname{Dom}\tilde{f} \to \Re$ and $\tilde{g} : \mathcal{Z} \to (-\infty, \infty]$ be functions such that:

F.1 $\tilde{g}$ is a proper closed convex function;

F.2 $\tilde{f}$ is differentiable and convex on a nonempty closed convex set $\Omega \supset \operatorname{dom}\tilde{g}$.

F.3 $\nabla\tilde{f}$ is $L_{\tilde{f}}$-Lipschitz continuous on Ω;

The generic problem mentioned above is as follows.

(P1) Given $u_0 \in \mathcal{Z}$, scalars $\lambda > 0$ and $\sigma_z \geq 0$, find a triple $(\tilde{u}, \tilde{s}, \tilde{\varepsilon})$ such that

$$\tilde{s} \in \partial_{\tilde{\varepsilon}} \tilde{g}(\tilde{u}), \quad \|\lambda(\nabla \tilde{f}(\tilde{u}) + \tilde{s}) + \tilde{u} - u_0\|^2 + 2\lambda\tilde{\varepsilon} \leq \sigma_z^2 \|\tilde{u} - u_0\|^2. \tag{85}$$

We now make a few remarks about (P1). First, steps 1 and 2 of the CSP-BD-HPE framework are clearly special cases of the above generic problem. Indeed, step 1 is a special case of (P1) in which $\tilde{f}$ is equal to the function $x \in \Omega_x \mapsto \Psi(x, y'_{k-1})$, $\tilde{g} = g_1$ and $\Omega = \Omega_x$, while step 2 is a special case of (P1) in which $\tilde{f}$ is equal to the function $y \in \Omega_y \mapsto \Psi(\tilde{x}_k, y)$, $\tilde{g} = g_2$ and $\Omega = \Omega_y$. Second, the above problem is related to the problem of finding an approximate solution of the (strongly) convex optimization problem

$$\min_{u \in \mathcal{Z}} \left\{ \frac{1}{2}\|u - u_0\|^2 + \lambda[\tilde{f}(u) + \tilde{g}(u)] \right\}. \tag{86}$$

Clearly, an exact solution $\tilde{u}$ of problem (86) satisfies $0 \in \tilde{u} - u_0 + \lambda(\nabla \tilde{f}(\tilde{u}) + \partial \tilde{g}(\tilde{u}))$, which implies that the triple $(\tilde{u}, \tilde{s}, \tilde{\varepsilon})$ where

$$\tilde{s} = \frac{u_0 - \tilde{u}}{\lambda} - \nabla \tilde{f}(\tilde{u}), \quad \tilde{\varepsilon} = 0 \tag{87}$$

satisfies (85) with $\sigma_z = 0$ (and hence, with any $\sigma_z \geq 0$). Hence, the situation in which (86) can be solved exactly immediately yields a solution of problem (P1).

In the rest of this chapter, we describe two instances of CSP-BD-HPE framework which differ in the way they solve the generic problem (P1). Section 4.2 states a specific instance of the CSP-BD-HPE framework in which λ is chosen sufficiently small, and steps 1 and 2, when viewed as problem (P1), are solved by performing a single composite gradient step on (86). Section 4.3 derives the iteration-complexity of solving problem (P1) by means of a Nesterov's accelerated variant. Section 4.4 describes an instance of the CSP-BD-HPE framework in which λ is chosen as the

maximum value allowed by the framework, i.e., the right hand side of (72), and the two triples in steps 1 and 2 of the CSP-BD-HPE framework are computed by means of the Nesterov's accelerated variant described in Section 4.3.

4.2 A special instance of the CSP-BD-HPE framework

This section states a specific instance of the CSP-BD-HPE framework in which steps 1 and 2 of the CSP-BD-HPE framework, when viewed as problem (P1), can be solved by performing a single composite gradient step on (86). The instance can be viewed as a block-decomposition version of Tseng's modified forward-backward splitting algorithm in [59] (see also [37, 35]). Except for our slightly more general way of choosing the stepsize, this instance is the same as the method stated in Subsection 5.2 of [36] when the latter one is specialized to the context of (70)-(71). Despite this similarity, we include a detailed discussion of the aforementioned instance in this section in order to motivate the accelerated instance and its corresponding complexity bounds presented in Section 4.4.

The following result, whose proof can be found in Proposition 4.3 of [36], shows that a single composite gradient step from u_0 with respect to (86) yields a solution of (P1) when $\nabla \tilde{f}$ is Lipschitz continuous on Ω, λ is sufficiently small and the resolvent of $\partial \tilde{g}$, i.e., a vector of the form $(I + \lambda \partial \tilde{g})^{-1}(w)$ for some $\lambda > 0$, can be computed at any point $w \in \mathcal{Z}$. Clearly,

$$(I + \lambda \partial \tilde{g})^{-1}(w) = \operatorname{argmin}_{u \in \mathcal{Z}} \left\{ \tilde{g}(u) + \frac{1}{2\lambda} \|u - w\|^2 \right\}.$$

Also, when $\tilde{g}$ is the indicator of a nonempty closed convex set $\Omega \subset \mathcal{Z}$, $(I + \lambda \tilde{g})^{-1}(\cdot) = P_\Omega(\cdot)$ for any $\lambda > 0$.

Proposition 4.2.1. *For some $\tilde{L} \geq 0$, assume that $\nabla \tilde{f}$ is $\tilde{L}$-Lipschitz continuous on Ω. Then, for any $u_0 \in \mathcal{Z}$, $\sigma_z \geq 0$ and $\lambda > 0$ such that $\lambda \tilde{L} \leq \sigma_z$, the triple $(\tilde{x}, \tilde{s}, \tilde{\varepsilon})$*

given by

$$\tilde{u} := (I+\lambda\partial\tilde{g})^{-1}\left(u_0 - \lambda\nabla\tilde{f}(P_\Omega(u_0))\right), \quad \tilde{s} := \frac{1}{\lambda}(u_0-\tilde{u})-\nabla\tilde{f}(P_\Omega(u_0)), \quad \tilde{\varepsilon} := 0 \tag{88}$$

solves problem (P1).

It is worth noting that when $\tilde{L} = 0$, i.e., $\tilde{f}$ is an affine function on Ω, the point $\tilde{u}$ in (88) is the exact solution of problem (86) and the triple $(\tilde{u}, \tilde{s}, \tilde{\varepsilon})$ given by (88) solves problem (P1) for any $\sigma_z \geq 0$. Hence, when $\tilde{L} = 0$, the recipe of Proposition 4.2.1 is equivalent to obtaining $\tilde{u}$ by simply solving problem (86) exactly and obtaining $(\tilde{s}, \tilde{\varepsilon})$ as in (87).

In the remaining part of this section, we will explicitly state a special case of the CSP-BD-HPE framework in which the triples in both steps 1 and 2 are computed by means of the recipe described in Proposition 4.2.1.

We now state a special case of the CSP-BD-HPE framework in which the two triple finding problems in steps 1 and 2 are implicitly solved using the recipe of Proposition 4.2.1.

[T-BD] Tseng's based BD-HPE algorithm for CSP$(\Psi; \mathbf{g_1}, \mathbf{g_2})$:

0) Let $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$ and $\sigma \in (0, 1]$ be given, choose $\sigma_x, \sigma_y \in [0, \sigma)$, and set $k = 1$ and

$$\lambda = \bar{\lambda}(\sigma_x, \sigma_y) := \min\left\{\frac{\sigma_x}{L_{xx}}, \frac{\sigma_y}{L_{yy}}, \frac{\sqrt{(\sigma^2-\sigma_x^2)(\sigma^2-\sigma_y^2)}}{\sigma L_{xy}}\right\} \tag{89}$$

(with the convention that $0/0 = \infty$);

1) compute $x'_{k-1} := P_{\Omega_x}(x_{k-1})$, $y'_{k-1} := P_{\Omega_y}(y_{k-1})$ and the pair $(\tilde{x}_k, \tilde{y}_k) \in \mathcal{X} \times \mathcal{Y}$ as

$$\tilde{x}_k := [I+\lambda\partial g_1]^{-1}(x_{k-1}-\lambda\nabla_x\Psi(x'_{k-1}, y'_{k-1})), \;\; \tilde{y}_k := [I+\lambda\partial g_2]^{-1}(y_{k-1}+\lambda\nabla_y\Psi(\tilde{x}_k, y'_{k-1})); \tag{90}$$

2) compute (x_k, y_k) as

$$x_k := \tilde{x}_k - \lambda[\nabla_x \Psi(\tilde{x}_k, \tilde{y}_k) - \nabla_x \Psi(x'_{k-1}, y'_{k-1})], \;\; y_k := \tilde{y}_k + \lambda[\nabla_y \Psi(\tilde{x}_k, \tilde{y}_k) - \nabla_y \Psi(\tilde{x}_k, y'_{k-1})], \tag{91}$$

set $k \leftarrow k + 1$, and go to step 1.

end

Note that an iteration of the T-BD algorithm requires two evaluations of each of the partial derivatives $\nabla_x \Psi(\cdot, \cdot)$ and $\nabla_y \Psi(\cdot, \cdot)$, one evaluation of each of the resolvents of ∂g_1 and ∂g_2 and one projection evaluation onto each one of the sets Ω_x and Ω_y. Hence, an iteration-complexity bound derived for the above algorithm is also a bound on the number of each of the above operations performed throughout the algorithm.

The following result, which follows as a consequence of Theorem 4.1.3, establishes the iteration-complexity of the T-BD algorithm. Even though we could prove it for a more specific choice of stepsize λ using Theorem 5.4 of [36], we have decided for the sake of completeness to include a self-contained but short proof here. This decision was also based on the fact that the complexity bounds obtained here uses the notion of approximate solution given in Definition 2.1.1 rather than the one used in [36].

Theorem 4.2.2. *Consider the sequences $\{(x_k, y_k)\}$ and $\{(\tilde{x}_k, \tilde{y}_k)\}$ generated by the T-BD algorithm, and define the sequences $\{(\tilde{r}_k^x, \tilde{r}_k^y)\}$ and $\{(\varepsilon_k^x, \varepsilon_k^y)\}$ as*

$$\tilde{r}_k^x := \frac{1}{\lambda}(x_{k-1} - x_k), \quad \tilde{r}_k^y := \frac{1}{\lambda}(y_{k-1} - y_k), \quad \varepsilon_k^x := 0, \quad \varepsilon_k^y := 0,$$

$\{(\tilde{x}_k^a, \tilde{y}_k^a)\}$ and $\{\tilde{r}_k^a\}$ as in (82), and $\{\tilde{\varepsilon}_k^a\}$ as in (83), and set

$$M := \max\{L_{xx}, L_{xy}, L_{yy}\}. \tag{92}$$

Then, for every pair of positive scalars (ρ, ε), there exists an index

$$k_0 = \mathcal{O}\left(M \max\left[\frac{d_0^2}{\varepsilon}, \frac{d_0}{\rho}\right]\right) \tag{93}$$

such that the average point $(\tilde{x}_k^a, \tilde{y}_k^a)$ *is a* (ρ, ε)*-saddle-point of* $CSP(\Psi; g_1, g_2)$ *for every* $k \geq k_0$.

Proof. We first show that the T-BD algorithm is a special case of the CSP-BD-HPE framework. Indeed, assumptions B.4 and B.5 together with Proposition 4.2.1 used twice, namely, first with $(\tilde{f}, \tilde{g}, u_0, \sigma_z) = (\Psi(\cdot, y'_{k-1}), g_1, x_{k-1}, \sigma_x)$ and then with $(\tilde{f}, \tilde{g}, u_0, \sigma_z) = (-\Psi(\tilde{x}_k, \cdot), g_2, y_{k-1}, \sigma_y)$, imply that the triples $(\tilde{x}_k, \tilde{a}_k, \varepsilon_k^x)$ and $(\tilde{y}_k, \tilde{b}_k, \varepsilon_k^y)$, where $\tilde{x}_k$ and $\tilde{y}_k$ are given by (90), $\varepsilon_k^x = \varepsilon_k^y := 0$ and

$$\tilde{a}_k := \frac{1}{\lambda}(x_{k-1} - \tilde{x}_k) - \nabla_x \Psi(x'_{k-1}, y'_{k-1}), \quad \tilde{b}_k := \frac{1}{\lambda}(y_{k-1} - \tilde{y}_k) + \nabla_y \Psi(\tilde{x}_k, y'_{k-1}),$$

satisfy (73) and (74). Now, using the above formulae for $\tilde{a}_k$ and $\tilde{b}_k$, we easily see that step 3 of CSP-BD-HPE reduces to the update formula (91). We have thus shown the above claim which, as a consequence, implies that the pair $(\tilde{r}_k^a, \tilde{\varepsilon}_k^a)$ satisfies the conclusions of Theorem 4.1.3(b). The conclusion of theorem then follows immediately from bound (84) and formula (89) for the stepsize λ. □

Given a fixed $\sigma \in (0, 1]$, it can be shown that choosing $\sigma_x = \bar{\lambda} L_{xx}$ and $\sigma_y = \bar{\lambda} L_{yy}$ where

$$\bar{\lambda} := \sigma \left\{ \theta_{\max} \left(\begin{bmatrix} L_{xx}^2 & L_{xx} L_{xy} \\ L_{xx} L_{xy} & L_{yy}^2 + L_{xy}^2 \end{bmatrix} \right) \right\}^{-1/2},$$

maximizes $\bar{\lambda}(\sigma_x, \sigma_y)$ under the condition that $\sigma_x, \sigma_y \in [0, \sigma)$, in which case $\bar{\lambda}(\sigma_x, \sigma_y) = \bar{\lambda}$. As a consequence, this choice of σ_x and σ_y minimizes the convergence rate bounds established in Theorem 4.1.3 for the CSP-BD-HPE framework, and hence the T-BD algorithm. We observe that the above choice of stepsize is the exactly the one used by the BD algorithm of Subsection 5.2 of [36].

It is worth emphasizing that the T-BD algorithm uses the recipe described in Proposition 4.2.1 in order to compute the triples in steps 1 and 2 of the CSP-BD-HPE framework. Hence, it is necessary to choose a stepsize $\lambda > 0$ which satisfies, in addition to (72), the first two inequalities in (89). As a result, the largest λ that can be

chosen in this manner, namely as in (89), is $1/\mathcal{O}(M)$. On the other hand, the largest stepsize λ that can be chosen in the context of the CSP-BD-HPE framework, i.e., as the right hand side of (72), is $1/\mathcal{O}(L_{xy})$. Clearly, when $M >> L_{xy}$, or equivalently $\max\{L_{xx}, L_{yy}\} >> L_{xy}$, the latter stepsize is considerably larger than the first one. As a consequence, the number of iterations performed by an arbitrary CSP-BD-HPE instance with λ equal to the right hand side of (72) can be considerably smaller than the number of iterations performed by the T-BD algorithm with λ chosen according to (89). Needless to say, instances of the CSP-BD-HPE framework with λ equal to the right hand side of (72) can not be implemented with the aid of the recipe described in Proposition 4.2.1 and requires instead a different approach.

4.3 An accelerated method for problem (P1)

The main goal of this section is to derive the iteration-complexity of solving problem (P1) using the variant of the Nesterov's accelerated method introduced in Section 3.1.

We start this section by describing a technical result.

Lemma 4.3.1. *Assume that f and h are functions such that h is a proper closed convex function, f is differentiable on a closed convex set $\Omega \supset \operatorname{dom} h$, and that there exists $L > 0$ such that ∇f is L-Lipschitz continuous on Ω. Then, if $(z, r, \varepsilon) \in \mathcal{Z} \times \mathcal{Z} \times \Re_+$ satisfy*

$$r \in \partial_\varepsilon (f + h)(z), \tag{94}$$

for any positive scalar $c > L/2$, the vector

$$\delta_c = \delta(z, r, c, f, h) := c[z - (I + c^{-1}\partial h)^{-1}(z - c^{-1}\nabla f(z) + c^{-1} r)] \tag{95}$$

satisfies

$$r + \delta_c \in (\nabla f + \partial_\varepsilon h)(z), \quad \|\delta_c\| \leq c\sqrt{\frac{2\varepsilon}{2c - L}}. \tag{96}$$

Proof. First note that it is well-known that conditions on f implies that

$$0 \leq f(u') - f(u) - \langle \nabla f(u), u' - u \rangle \leq \frac{L}{2}\|u' - u\|^2 \quad \forall u, u' \in \Omega. \tag{97}$$

Note also that inclusion (94) implies that $z \in \operatorname{dom} h \subset \Omega$. It is easy to see that (95) implies that

$$z - c^{-1}\delta_c \in \operatorname{dom} h, \quad r + \delta_c - \nabla f(z) \in \partial h(z - c^{-1}\delta_c).$$

The above inclusion implies

$$h(u) - h(z - c^{-1}\delta_c) \geq \langle r + \delta_c - \nabla f(z), u - z + c^{-1}\delta_c \rangle \quad \forall u \in \mathcal{Z}. \tag{98}$$

On the other hand, it follows from (94) that

$$f(u) + h(u) \geq f(z) + h(z) + \langle r, u - z \rangle - \varepsilon \quad \forall u \in \mathcal{Z}.$$

This inequality with $u = z - c^{-1}\delta_c$ and (97) with $u' = z$ and $u = z - c^{-1}\delta$ then imply that

$$\begin{aligned} h(z - c^{-1}\delta_c) - h(z) &\geq -[f(z - c^{-1}\delta_c) - f(z) - \langle \nabla f(z), -c^{-1}\delta_c \rangle] + \langle r - \nabla f(z), -c^{-1}\delta_c \rangle - \varepsilon \\ &\geq -\frac{L}{2}\|c^{-1}\delta_c\|^2 + \langle r - \nabla f(z), -c^{-1}\delta_c \rangle - \varepsilon. \end{aligned} \tag{99}$$

Adding up (98) and (99), we conclude that

$$h(u) - h(z) \geq \langle r + \delta_c - \nabla f(z), u - z \rangle + c^{-2}\left(c - \frac{L}{2}\right)\|\delta_c\|^2 - \varepsilon. \quad \forall u \in \mathcal{Z},$$

which clearly implies the inclusion in (96) when $c \geq L/2$. Moreover, this same inequality with $u = z$ implies the inequality in (96) when $c > L/2$. □

Note that the above lemma described a procedure which finds a triple satisfying the inclusion in (96) based on a triple satisfying the weaker inclusion in (94). Therefore, the lemma is useful in the construction of the following algorithm which aims at finding the solution of problem (P1).

We now state a variant of Nesterov's accelerated algorithm for solving problem (P1) which is based on the version of Algorithm 0 in Section 3.1 with $\psi(u) = \lambda\tilde{f}(u)$ and $g(u) = \lambda\tilde{g}(u) + \frac{1}{2}\|u - u_0\|^2$ and the procedure described in Lemma 4.3.1.

[Algorithm 1] A variant of Nesterov's accelerated algorithm for (P1):

0) Let $u_0 \in \mathcal{X}$ be given and set $\Gamma_0 = 0$, $\tilde{u}_0 = w_0 = P_\Omega(u_0)$, $k = 1$;

1) let $\Gamma_k > \Gamma_{k-1}$ be such that

$$\Gamma_k(\Gamma_{k-1} + 1) = L_{\tilde{f}}(\Gamma_k - \Gamma_{k-1})^2 \tag{100}$$

and compute $(u_k, w_k, \tilde{u}_k) \in \Omega \times \operatorname{dom}\tilde{g} \times \operatorname{dom}\tilde{g}$ as

$$u_k := \frac{\Gamma_{k-1}}{\Gamma_k}\tilde{u}_{k-1} + \frac{\Gamma_k - \Gamma_{k-1}}{\Gamma_k}w_{k-1}, \tag{101}$$

$$w_k := \operatorname{argmin} \sum_{i=1}^{k} \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma_k} l_{\lambda\tilde{f}(u)}(u; u_i) + \lambda\tilde{g}(u) + \frac{\Gamma_k + 1}{2\Gamma_k}\|u - u_0\|^2, \tag{102}$$

$$\tilde{u}_k := \frac{\Gamma_{k-1}}{\Gamma_k}\tilde{u}_{k-1} + \frac{\Gamma_k - \Gamma_{k-1}}{\Gamma_k}w_k; \tag{103}$$

2) compute

$$r_k := \frac{1}{\lambda\Gamma_k}(u_0 - w_k), \quad \delta_k := \delta(\tilde{u}_k, r_k, L_{\tilde{f}} + 1/\lambda, \tilde{f} + \frac{1}{2\lambda}\|\cdot - u_0\|^2, \tilde{g}), \tag{104}$$

$$\tilde{\varepsilon}_k := \frac{1}{2\lambda\Gamma_k}\|\tilde{u}_k - u_0\|^2 - \frac{1}{2\lambda\Gamma_k}\|\tilde{u}_k - w_k\|^2, \tag{105}$$

$$\tilde{s}_k := r_k + \delta_k - \frac{(\tilde{u}_k - u_0)}{\lambda} - \nabla\tilde{f}(\tilde{u}_k), \tag{106}$$

where $\delta(\cdot,\cdot,\cdot,\cdot,\cdot)$ is defined in (95);

3) set $k \leftarrow k + 1$ and go to step 1.

end

We now derive the iteration-complexity of Algorithm 2 for solving problem (P1).

Proposition 4.3.2. *Consider the sequences $\{\tilde{u}_k\}$, $\{r_k\}$ and $\{\tilde{\varepsilon}_k\}$ generated by Algorithm 1. Then, for any $u_0 \in \mathcal{Z}$ and $\tau > 0$, there exists an index*

$$k_0 = \mathcal{O}\left(\left\lceil \min\left\{\sqrt{\lambda L_{\tilde{f}}\lceil \tau^{-1}\rceil},\, 1 + \left(1+\sqrt{\lambda L_{\tilde{f}}}\right)\log^+\left(\lambda L_{\tilde{f}}\lceil \tau^{-1}\rceil\right)\right\}\right\rceil\right) \tag{107}$$

such that, for every $k \geq k_0$, the triple $(\tilde{u}, r, \tilde{\varepsilon}) = (\tilde{u}_k, r_k, \tilde{\varepsilon}_k)$ satisfies

$$r \in \partial_{\tilde{\varepsilon}}\left(\tilde{f} + \tilde{g} + \frac{1}{2\lambda}\|\cdot - u_0\|^2\right)(\tilde{u}), \tag{108}$$

$$\|\lambda r\|^2 + 2\lambda\tilde{\varepsilon} \leq \tau\|\tilde{u} - u_0\|^2. \tag{109}$$

Proof. We first claim that for every $k \geq 1$ such that

$$\Gamma_k \geq \max\{2, 2\tau^{-1}\}, \tag{110}$$

the triple $(\tilde{u}, r, \tilde{\varepsilon}) = (\tilde{u}_k, r_k, \tilde{\varepsilon}_k)$ satisfies (108) and (109). Indeed, it follows from (44) with $p := \lambda\tilde{f} + \lambda\tilde{g} + \frac{1}{2}\|\cdot - u_0\|^2$ that for any $u \in \mathcal{Z}$ and $k \geq 1$,

$$\begin{aligned} p(u) - p(\tilde{u}_k) &\geq \frac{1}{2\Gamma_k}\left(\|u - w_k\|^2 - \|u - u_0\|^2\right) \\ &= \frac{1}{\Gamma_k}\langle u_0 - w_k, u - \tilde{u}_k\rangle - \frac{1}{2\Gamma_k}(\|\tilde{u}_k - u_0\|^2 - \|\tilde{u}_k - w_k\|^2). \end{aligned}$$

The above inequality together with definitions of r_k and $\tilde{\varepsilon}_k$ in (104) and (105) and the definition of ε-subdifferential in (12) then imply that $\lambda r_k \in \partial_{\lambda\tilde{\varepsilon}_k} p(\tilde{u}_k)$ and hence the triple $(\tilde{u}, r, \tilde{\varepsilon}) = (\tilde{u}_k, r_k, \tilde{\varepsilon}_k)$ satisfies inclusion (108) for every $k \geq 1$. Moreover, by (104) and the inequality $\|a+b\|^2 \leq 2(\|a\|^2 + \|b\|^2)$, we have

$$\|\lambda r_k\|^2 = \frac{1}{\Gamma_k^2}\|w_k - u_0\|^2 \leq \frac{2}{\Gamma_k^2}\|\tilde{u}_k - u_0\|^2 + \frac{2}{\Gamma_k^2}\|\tilde{u}_k - w_k\|^2,$$

and hence that

$$\|\lambda r_k\|^2 + 2\lambda\tilde{\varepsilon}_k \leq \left(\frac{2}{\Gamma_k^2} + \frac{1}{\Gamma_k}\right)\|\tilde{u}_k - u_0\|^2 + \left(\frac{2}{\Gamma_k^2} - \frac{1}{\Gamma_k}\right)\|\tilde{u}_k - w_k\|^2. \tag{111}$$

Since condition (110) is easily seen to imply that

$$\frac{2}{\Gamma_k^2} + \frac{1}{\Gamma_k} \leq \tau, \quad \frac{2}{\Gamma_k^2} - \frac{1}{\Gamma_k} \leq 0,$$

we conclude from (111) that the triple $(\tilde{u}, r, \tilde{\varepsilon}) = (\tilde{u}_k, r_k, \tilde{\varepsilon}_k)$ satisfies (109) for every $k \geq 1$ satisfying (110). We have thus shown that the above claim holds.

Now, define

$$k_0 := \left\lceil \min \left\{ 2\sqrt{\lambda L_{\tilde{f}} \lceil \tau^{-1} \rceil}\,,\; 1 + \frac{1 + \sqrt{1/(2\lambda L_{\tilde{f}})}}{2\sqrt{1/(2\lambda L_{\tilde{f}})}} \log^+ \left(\lambda L_{\tilde{f}} \lceil \tau^{-1} \rceil \right) \right\} \right\rceil \tag{112}$$

and note that k_0 clearly satisfies (107) and $k_0 \geq 1$. To end the proof, it suffices to show in view of the above claim that $k \geq k_0$ implies (110). Indeed, in view of the inequality $t/(1+t) \leq \log(1+t)$ for $t > -1$, we have

$$\frac{1 + \sqrt{1/(2\lambda L_{\tilde{f}})}}{\sqrt{1/(2\lambda L_{\tilde{f}})}} \geq \frac{1}{\log(1 + \sqrt{1/(2\lambda L_{\tilde{f}})})},$$

Thus, $k \geq k_0$ implies that either

$$k \geq 2\sqrt{\lambda L_{\tilde{f}} \lceil \tau^{-1} \rceil} \quad \text{or} \quad k \geq 1 + \frac{\log \left(\lambda L_{\tilde{f}} \lceil \tau^{-1} \rceil \right)}{2 \log(1 + \sqrt{1/(2\lambda L_{\tilde{f}})})}$$

and hence that

$$\Gamma_k \geq \max \left\{ \frac{k^2}{2\lambda L_{\tilde{f}}}\,,\; \frac{2}{\lambda L_{\tilde{f}}} \left(1 + \sqrt{\frac{1}{2\lambda L_{\tilde{f}}}} \right)^{2(k-1)} \right\} \geq 2 \left\lceil \tau^{-1} \right\rceil \geq \max\{2, 2\tau^{-1}\}.$$

where the first inequality is due to (43) and the fact that $\psi = \lambda \tilde{f}$ and $g = \lambda \tilde{g} + \frac{1}{2}\| \cdot - u_0\|^2$ satisfy conditions D.3 and D.4 with $L = \lambda L_{\tilde{f}}$ and $\mu = 1$. □

We are now ready to establish the iteration-complexity of Algorithm 2 for solving (P1), which is a consequence of Proposition 4.3.2 and Lemma 4.3.1.

Corollary 4.3.3. *For a given triple $(u_0, \lambda, \sigma_z) \in \mathcal{Z} \times \Re_{++} \times (0, 1]$, consider the sequences $\{\tilde{u}_k\}$, $\{\tilde{s}_k\}$ and $\{\tilde{\varepsilon}_k\}$ generated by Algorithm 1. Then, there exists an index*

$$k_0 = \mathcal{O}\left(1 + \sqrt{\lambda L_{\tilde{f}} + 1} \log \left((\lambda L_{\tilde{f}} + 1)/\sigma_z \right) \right) \tag{113}$$

such that the triple $(\tilde{u}, \tilde{s}, \tilde{\varepsilon}) = (\tilde{u}_k, \tilde{s}_k, \tilde{\varepsilon}_k)$ is a solution of problem (P1) for every $k \geq k_0$.

Proof. Let $\tau = \sigma_z^2/[2(\lambda L_{\tilde{f}} + 2)]$. In view of Proposition 4.3.2, there exists an index such that (107) holds and for every the triple $(\tilde{u}, r, \tilde{\varepsilon}) = (\tilde{u}_k, r_k, \tilde{\varepsilon}_k)$ satisfies (108) and (109) for any $k \geq k_0$. Now use the fact that $\lceil \tau^{-1} \rceil \leq 2(\lambda L_{\tilde{f}} + 1)\lceil \sigma_z^{-2} \rceil$, it is easy to see that (107) implies

$$k_0 = \mathcal{O}\left(\min\left\{(\lambda L_{\tilde{f}} + 2)\sigma_z^{-1}\,,\, 1 + \sqrt{(\lambda L_{\tilde{f}} + 2)}\,\log\left((\lambda L_{\tilde{f}} + 2)\sigma_z^{-1}\right)\right\}\right), \qquad (114)$$

which implies (113) in view of the assumption that $\sigma_z \in (0, 1]$ and the inequality $2t \geq \sqrt{t}\log t$ for $t > 0$.

In view of Lemma 4.3.1 with $f = \tilde{f} + \frac{1}{2\lambda}\|\cdot - u_0\|^2$, $h = \tilde{g}$ and $c = L_{\tilde{f}} + 1/\lambda$, and the fact $(\tilde{u}, r, \tilde{\varepsilon}) = (\tilde{u}_k, r_k, \tilde{\varepsilon}_k)$ satisfies (108) for any $k \geq k_0$, we have

$$r_k + \delta_k \in \nabla\tilde{f}(\tilde{u}_k) + \frac{(\tilde{u}_k - u_0)}{\lambda} + \partial_{\tilde{\varepsilon}_k}\tilde{g}(\tilde{u}_k), \quad \|\delta_k\|^2 \leq 2(L_{\tilde{f}} + 1/\lambda)\tilde{\varepsilon}_k.$$

It is easy to see the above inclusion and the definition of $\tilde{s}_k$ in (106) imply $(\tilde{u}, \tilde{s}, \tilde{\varepsilon}) = (\tilde{u}_k, \tilde{s}_k, \tilde{\varepsilon}_k)$ satisfies the inclusion in (85). Moreover, the above inequality, together with the definition of $\tilde{s}_k$ in (106), $\tau = \sigma_z^2/[2(\lambda L_{\tilde{f}}+2)]$ and the fact $(\tilde{u}, r, \tilde{\varepsilon}) = (\tilde{u}_k, r_k, \tilde{\varepsilon}_k)$ satisfies (109), implies that

$$\begin{aligned}
&\|\lambda(\nabla\tilde{f}(\tilde{u}_k) + \tilde{s}_k) + \tilde{u}_k - u_0\|^2 + 2\lambda\tilde{\varepsilon}_k = \|\lambda r_k + \lambda\delta_k\|^2 + 2\lambda\tilde{\varepsilon}_k \\
&\leq 2(\lambda L_{\tilde{f}} + 2)(\|\lambda r_k\|^2 + 2\lambda\tilde{\varepsilon}_k) \leq \sigma_z^2\|\tilde{u}_k - u_0\|^2.
\end{aligned}$$

□

Note that, when $\sigma_z \in (0, 1]$ is such that $\sigma_z^{-1} = \mathcal{O}(1)$, the iteration-complexity of solving problem (P1) by means of Algorithm 1 reduces to

$$\mathcal{O}\left(1 + \sqrt{\lambda L_{\tilde{f}} + 1}\,\log\left(\lambda L_{\tilde{f}} + 1\right)\right).$$

4.4 Accelerated BD Algorithm for CSP Problem

This section considers a special instance of the CSP-BD-HPE framework for solving the CSP problem $CSP(\Psi; g_1, g_2)$ in which the triples $(\tilde{x}_k, \tilde{a}_k, \varepsilon_k^x)$ and $(\tilde{y}_k, \tilde{b}_k, \varepsilon_k^y)$ in

steps 1 and 2 are obtained with the aid of Algorithm 1 applied to specific instances of problem (P1). It also establishes the complexity of the resulting accelerated instance in terms of gradient, projection and resolvent evaluations, and shows that it is substantially better than that of the Tseng's based BD-HPE algorithm when $\max\{L_{xx}, L_{yy}\} >> L_{xy}$.

We now state the special instance of the CSP-BD-HPE framework for solving $CSP(\Psi; g_1, g_2)$.

[Acc-BD] an accelerated BD-HPE algorithm for $CSP(\Psi; g_1, g_2)$:

0) Let $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$, $\sigma \in (0, 1]$, and $\sigma_x, \sigma_y \in (0, \sigma)$ be given. Set $k = 1$ and

$$\lambda = \frac{\sqrt{(\sigma^2 - \sigma_x^2)(\sigma^2 - \sigma_y^2)}}{\sigma L_{xy}}; \tag{115}$$

1) invoke Algorithm 1 with $u_0 = x_{k-1}$, $\mathcal{Z} = \mathcal{X}$,

$$\Omega = \Omega_x, \quad \tilde{g}(\cdot) = g_1(\cdot), \quad \tilde{f}(\cdot) = \Psi(\cdot, y'_{k-1}),$$

where $y'_{k-1} = P_{\Omega_y}(y_{k-1})$, to obtain a triple $(\tilde{u}, \tilde{s}, \tilde{\varepsilon}) \in \mathcal{X} \times \mathcal{X} \times \Re_+$ as in (103), (106) and (105) and set $(\tilde{x}_k, \tilde{a}_k, \varepsilon_k^x) = (\tilde{u}, \tilde{s}, \tilde{\varepsilon})$;

2) invoke Algorithm 1 with $w_0 = y_{k-1}$, $\mathcal{Z} = \mathcal{Y}$,

$$\Omega = \Omega_y, \quad \tilde{g}(\cdot) = g_2(\cdot), \quad \tilde{f}(\cdot) = -\Psi(\tilde{x}_k, \cdot)$$

to obtain a triple $(\tilde{u}, \tilde{s}, \tilde{\varepsilon}) \in \mathcal{Y} \times \mathcal{Y} \times \Re_+$ as in (103), (106) and (105) and set $(\tilde{y}_k, \tilde{b}_k, \varepsilon_k^y) = (\tilde{u}, \tilde{s}, \tilde{\varepsilon})$;

3) set $(\tilde{r}_k^x, \tilde{r}_k^y) = (\nabla_x \Psi(\tilde{x}_k, \tilde{y}_k) + \tilde{a}_k, -\nabla_y \Psi(\tilde{x}_k, \tilde{y}_k) + \tilde{b}_k)$,

$$(x_k, y_k) = (x_{k-1}, y_{k-1}) - \lambda(\tilde{r}_k^x, \tilde{r}_k^y), \tag{116}$$

and $k \leftarrow k + 1$, and go to step 1).

end

The following result establishes the iteration-complexity result of Acc-BD algorithm. The bounds derived on it are obtained under the assumption that the parameters σ, σ_x and σ_y are chosen so that the inverses of σ_x, σ_y, $\sigma^2 - \sigma_x^2$ and $\sigma^2 - \sigma_y^2$ are all $\mathcal{O}(1)$.

Theorem 4.4.1. *Algorithm Acc-BD is a special instance of the CSP-BD-HPE framework for solving $CSP(\Psi; g_1, g_2)$. Moreover, consider the sequences $\{(x_k, y_k)\}$, $\{(\tilde{x}_k, \tilde{y}_k)\}$, $\{(\varepsilon_k^x, \varepsilon_k^y)\}$ and $\{(\tilde{r}_k^x, \tilde{r}_k^y)\}$ generated by Acc-BD algorithm and define $\{(\tilde{x}_k^a, \tilde{y}_k^a)\}$, $\{\tilde{r}_k^a\}$, $\{\tilde{\varepsilon}_k^a\}$, d_0 and $\bar{\eta}$ as in Theorem 4.1.3. Then, the following statements hold:*

(a) for every $k \in \mathbb{N}$, $((\tilde{r}_k^x, \tilde{r}_k^y), \varepsilon_k^x + \varepsilon_k^y)$ is a SP-residual for $(\tilde{x}_k, \tilde{y}_k)$ with respect to $CSP(\Psi; g_1, g_2)$, or equivalently,

$$(\tilde{r}_k^x, \tilde{r}_k^y) \in \partial_{\varepsilon_k^x + \varepsilon_k^y}[\Psi(\cdot, \tilde{y}_k) + \Psi(\tilde{x}_k, \cdot)](\tilde{x}_k, \tilde{y}_k)$$

and there exists $i \leq k$ such that

$$\|(\tilde{r}_i^x, \tilde{r}_i^y)\| \leq \frac{L_{xy}\sigma d_0}{\sqrt{(\sigma^2 - \sigma_x^2)(\sigma^2 - \sigma_y^2)}}\sqrt{\frac{1+\sigma}{k(1-\sigma)}},$$
$$\varepsilon_i^x + \varepsilon_i^y \leq \frac{L_{xy}\sigma^3 d_0^2}{2k(1-\sigma^2)\sqrt{(\sigma^2 - \sigma_x^2)(\sigma^2 - \sigma_y^2)}},$$

(b) for every $k \in \mathbb{N}$, the pair $(\tilde{r}_k^a, \tilde{\varepsilon}_k^a)$ is a SP-residual of $(\tilde{x}_k^a, \tilde{y}_k^a)$ for $CSP(\Psi; g_1, g_2)$, or equivalently,

$$\tilde{r}_k^a \in \partial_{\tilde{\varepsilon}_k^a}[\Psi(\cdot, \tilde{y}_k^a) + \Psi(\tilde{x}_k^a, \cdot)](\tilde{x}_k^a, \tilde{y}_k^a)$$

and

$$\|\tilde{r}_k^a\| \leq \frac{2L_{xy}\sigma d_0}{k\sqrt{(\sigma^2 - \sigma_x^2)(\sigma^2 - \sigma_y^2)}}, \qquad \tilde{\varepsilon}_k^a \leq \frac{2L_{xy}\sigma d_0^2}{k\sqrt{(\sigma^2 - \sigma_x^2)(\sigma^2 - \sigma_y^2)}}(1 + \bar{\eta});$$

Proof. Acc-BD algorithm is clearly a special case of the CSP-BD-HPE framework in which (72) holds as equality and the triples of steps 1 and 2 are found by means of

Algorithm 1. (a) This statement follows immediately from Theorem 4.1.3(a) with λ as in (115). (b) This statement follows immediately from Theorem 4.1.3(b) with λ as in (115). $\square$

The following corollary follows immediately from (5.3.1), Corollary 4.3.3 and the fact that each iteration of Algorithm 1 performs at most two gradient evaluations, two resolvent evaluations of ∂h and one projection onto Ω.

Corollary 4.4.2. *At each iteration of Acc-BD algorithm, the number of evaluations of $\nabla_x \Psi(\cdot,\cdot)$ and the number of resolvent evaluations of ∂g_1 and $\partial \mathcal{I}_{\Omega_x}$ are both bounded by*

$$\mathcal{O}\left(1 + \sqrt{L_{xx}/L_{xy} + 1}\, \log(L_{xx}/L_{xy} + 1)\right);$$

and the number of evaluations of $\nabla_y \Psi(\cdot,\cdot)$ and the number of resolvent evaluations of ∂g_2 and $\partial \mathcal{I}_{\Omega_y}$ are bounded by

$$\mathcal{O}\left(1 + \sqrt{L_{yy}/L_{xy} + 1}\, \log(L_{yy}/L_{xy} + 1)\right).$$

As a consequence, for every pair of positive scalars (ρ, ε), Acc-BD algorithm finds a (ρ, ε)-saddle-point of $CSP(\Psi; g_1, g_2)$ by performing no more than

$$\mathcal{O}\left(\left[1 + \sqrt{L_{xx}/L_{xy} + 1}\, \log(L_{xx}/L_{xy} + 1)\right] \max\left\{\frac{L_{xy} d_0^2}{\varepsilon}\,,\, \frac{L_{xy} d_0}{\rho}\right\}\right) \tag{117}$$

evaluations of $\nabla_x \Psi(\cdot,\cdot)$ and resolvent evaluations of ∂g_1 and $\partial \mathcal{I}_{\Omega_x}$, and no more than

$$\mathcal{O}\left(\left[1 + \sqrt{L_{yy}/L_{xy} + 1}\, \log(L_{yy}/L_{xy} + 1)\right] \max\left\{\frac{L_{xy} d_0^2}{\varepsilon}\,,\, \frac{L_{xy} d_0}{\rho}\right\}\right) \tag{118}$$

evaluations of $\nabla_y \Psi(\cdot,\cdot)$ and resolvent evaluations of ∂g_2 and $\partial \mathcal{I}_{\Omega_y}$.

It is worthwhile to compare the iteration-complexity bound (93) obtained for the T-BD-HPE algorithm in Theorem 4.2.2 with the bounds (117) and (118) obtained for Acc-BD algorithm in Corollary 4.4.2. Indeed, when $\max\{L_{xx}, L_{yy}\} \approx L_{xy}$, the two bounds in (117) and (118) are of the same order of magnitude as the one in (93).

Consider now the relevant case in which $\max\{L_{xx}, L_{yy}\} >> L_{xy}$ and for the sake of concreteness that $L_{xx} = \max\{L_{xx}, L_{yy}\}$. Then, bound (93) is larger than (117) by a factor of

$$\Theta(\sqrt{\xi_x}/\log(\xi_x))$$

where $\xi_x := L_{xx}/L_{xy} >> 1$. Also, the bound (93) is significantly larger than (118), i.e., by a factor τ such that

$$\tau = \begin{cases} \Theta(\xi_x), & \text{when } L_{yy} = \mathcal{O}(L_{xy}) \\ \Theta(\xi_x/[\log(\xi_y)\sqrt{\xi_y}]), & \text{when } L_{yy} >> L_{xy} \end{cases},$$

where $\xi_y := L_{yy}/L_{xy}$.

In Subsection 6.1, we describe a relevant class of convex optimization problems corresponding to saddle-point problems with $L_{yy} = 0$ and the ratio chosen $\xi_x := L_{xx}/L_{xy}$ arbitrarily large. Moreover, the resolvent evaluations of ∂g_2 are much more expensive than the ones for ∂g_1. Note that this class of problems is particularly suitable for Acc-BD in view of the fact that the bound (118) on the number of expensive resolvent evaluations of ∂g_2 is significantly smaller than the bound (117) on the number of cheap resolvent evaluations of ∂g_1.

CHAPTER V

ACCELERATED HPE METHOD FOR A SPECIAL CLASS OF COMPOSITE SADDLE-POINT PROBLEMS

5.1 HPE framework for saddle-point problem

The section specializes the HPE framework to the context of the saddle-point problem and states its convergence properties.

We now state a special case of the HPE framework for solving the monotone inclusion problem (17), and hence the saddle-point problem $SP(\widehat{\Psi}; X, Y)$.

[SP-HPE] Hybrid proximal extragradient framework for solving $SP(\widehat{\Psi}; X, Y)$:

0) Let $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$, $\lambda > 0$ and $0 \leq \sigma < 1$ be given and set $k = 1$;

1) find $(\tilde{x}_k, \tilde{y}_k) \in \mathcal{X} \times \mathcal{Y}$, $\tilde{r}_k = (\tilde{r}_k^x, \tilde{r}_k^y) \in \mathcal{X} \times \mathcal{Y}$ and $\varepsilon_k \geq 0$ such that

$$(\tilde{r}_k^x, \tilde{r}_k^y) \in \partial_{\varepsilon_k}[\widehat{\Psi}(\cdot, \tilde{y}_k) - \widehat{\Psi}(\tilde{x}_k, \cdot)](\tilde{x}_k, \tilde{y}_k), \tag{119}$$

$$\begin{aligned} &\|\lambda\tilde{r}_k^x + \tilde{x}_k - x_{k-1}\|_{\mathcal{X}}^2 + \|\lambda\tilde{r}_k^y + \tilde{y}_k - y_{k-1}\|_{\mathcal{Y}}^2 + 2\lambda\varepsilon_k \\ &\leq \sigma^2 \left(\|\tilde{x}_k - x_{k-1}\|_{\mathcal{X}}^2 + \|\tilde{y}_k - y_{k-1}\|_{\mathcal{Y}}^2\right); \end{aligned} \tag{120}$$

2) set $x_k = x_{k-1} - \lambda\tilde{r}_k^x$, $y_k = y_{k-1} - \lambda\tilde{r}_k^y$ and $k \leftarrow k + 1$, and go to step 1.

end

We now make several remarks about the SP-HPE framework. First, due to Lemma 5.1.1 below, the SP-HPE framework is a special case of the HPE framework in which $\lambda_k := \lambda$. In fact, the SP-HPE framework could be stated in terms of a sequence of variable stepsizes $\{\lambda_k\}$, but we assume for simplicity $\lambda_k = \lambda$. Second, similar to the HPE framework, the SP-HPE framework does not specify how to find $(\tilde{x}_k, \tilde{y}_k)$, $\tilde{r}_k$

and ε_k satisfying the HPE error condition in (119) and (120). Section 5.3 describes a special instance of the SP-HPE framework in which $(\tilde{x}_k, \tilde{y}_k)$, $\tilde{r}_k$ and ε_k are obtained by a variant of Nesterov's accelerated method. Third, using the fact that the inclusion (119) is stronger than the inclusion in (54), we derive in Theorem 5.1.2 a finer version of Theorem 3.2.1 with $\lambda_k = \lambda$ specialized to the context of the saddle-point problem (16).

Before stating the pointwise and ergodic convergence rate results for the SP-HPE framework, we give two preliminary technical results.

Lemma 5.1.1. *For each $(x, y) \in X \times Y$ and $\varepsilon \geq 0$, we have*

$$\partial_\varepsilon(\widehat{\Psi}(\cdot, y) - \widehat{\Psi}(x, \cdot))(x, y) \subseteq T^\varepsilon(x, y),$$

where T is defined in (17)*.*

Proof. Let $r \in \partial_\varepsilon(\widehat{\Psi}(\cdot, y) - \widehat{\Psi}(x, \cdot))(x, y)$ be given. This clearly implies that

$$\widehat{\Psi}(\tilde{x}, y) - \widehat{\Psi}(x, \tilde{y}) \geq \langle (\tilde{x} - x, \tilde{y} - y), r\rangle - \varepsilon \qquad \forall (\tilde{x}, \tilde{y}) \in \mathcal{X} \times \mathcal{Y}.$$

On the other hand, it follows from the definition of T in (17) that any $\tilde{r} \in T(\tilde{x}, \tilde{y})$ satisfies

$$\widehat{\Psi}(x, \tilde{y}) - \widehat{\Psi}(\tilde{x}, y) \geq \langle (x - \tilde{x}, y - \tilde{y}), \tilde{r}\rangle.$$

Summing up the above two inequalities, we then conclude that

$$\langle (x - \tilde{x}, y - \tilde{y}), r - \tilde{r}\rangle \geq -\varepsilon \qquad \forall (\tilde{x}, \tilde{y}) \in \mathcal{X} \times \mathcal{Y},\ \forall \tilde{r} \in T(\tilde{x}, \tilde{y}),$$

and hence that $r \in T^\varepsilon(x, y)$ in view of the the definition of $T^\varepsilon(\cdot)$ in (10). □

The following result describes the pointwise and ergodic convergence rate properties of the SP-HPE framework.

Theorem 5.1.2. *Consider the sequences $\{(\tilde{x}_k, \tilde{y}_k)\}$,$\{(\tilde{r}_k^x, \tilde{r}_k^y)\}$ and $\{\varepsilon_k\}$ generated by the SP-HPE framework and define $(\tilde{x}_k^a, \tilde{y}_k^a), \tilde{r}_k^a$ and ε_k^a for every $k \in \mathbb{N}$ as in* (82) *and*

(83). *Let d_0 denote the distance of (x_0, y_0) to the solution set of $SP(\widehat{\Psi}; X, Y)$. Then, for every $k \in \mathbb{N}$, the following statements hold:*

(a) (pointwise convergence rate) the triple $((\tilde{x}_k, \tilde{y}_k), \tilde{r}_k, \varepsilon_k)$ is a $(\|\tilde{r}_k\|, \varepsilon_k)$-saddle-point of $\widehat{\Psi}$, or equivalently (119) *holds, and there exists an index $i \leq k$ such that*

$$\|\tilde{r}_i\| \leq \frac{d_0}{\lambda}\sqrt{\frac{1+\sigma}{k(1-\sigma)}}, \qquad \varepsilon_i \leq \frac{\sigma^2 d_0^2}{2k\lambda(1-\sigma^2)}; \tag{121}$$

(b) (ergodic convergence rate) the triple $((\tilde{x}_k^a, \tilde{y}_k^a), \tilde{r}_k^a, \varepsilon_k^a)$ is a $(\|\tilde{r}_k^a\|, \varepsilon_k^a)$-saddle-point of $\widehat{\Psi}$, or equivalently

$$\tilde{r}_k^a \in \partial_{\varepsilon_k^a}(\widehat{\Psi}(\cdot, \tilde{y}_k^a) - \widehat{\Psi}(\tilde{x}_k^a, \cdot))(\tilde{x}_k^a, \tilde{y}_k^a), \tag{122}$$

and

$$\|\tilde{r}_k^a\| \leq \frac{2d_0}{\lambda k}, \qquad 0 \leq \varepsilon_k^a \leq \frac{2d_0^2}{\lambda k}\left(1 + \frac{\sigma}{\sqrt{(1-\sigma^2)}}\right). \tag{123}$$

Proof. The first claim in (a) is obvious. Since, by (119) and Lemma 5.1.1, we have $\tilde{r}_k \in T^{\varepsilon_k}(\tilde{x}_k, \tilde{y}_k)$ where T is defined in (17), we conclude that the SP-HPE framework is a special instance of the HPE framework applied to (17) where $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ is endowed with the inner product defined in (14). The second claim in (a) then follows Theorem 3.2.1(a). Moreover, inclusion (122) follows from (119) and Lemma 4.1.2, and the bounds in (123) follow from Theorem 3.2.1(b) with $\lambda_k = \lambda$. □

5.2 Solving the HPE error condition

This section presents a scheme, together with its iteration-complexity analysis, for finding a solution of the HPE error condition (119)-(120) with $\widehat{\Psi}$ given by (6) (and w.l.o.g. $\lambda = 1$). The scheme is based on the Nesterov's accelerated variant of Subsection 3.1 applied to an associated composite saddle-point problem.

This section considers the following problem corresponding to the special case of step 1 of the SP-HPE framework in which $\lambda = 1$.

($P2$) Given convex sets $X \subset \mathcal{X}$ and $Y \subset \mathcal{Y}$, a closed convex-concave function $\widehat{\Psi}$ on $X \times Y$, a pair $(u_0, v_0) \in \mathcal{X} \times \mathcal{Y}$ and a scalar $\sigma > 0$, the problem is to find $(\tilde{u}, \tilde{v}) \in \mathcal{X} \times \mathcal{Y}, (\tilde{r}^u, \tilde{r}^v) \in \mathcal{X} \times \mathcal{Y}$ and $\tilde{\varepsilon} \geq 0$ such that

$$(\tilde{r}^u, \tilde{r}^v) \in \partial_{\tilde{\varepsilon}} \left[\widehat{\Psi}(\cdot, \tilde{v}) - \widehat{\Psi}(\tilde{u}, \cdot) \right] (\tilde{u}, \tilde{v}), \tag{124}$$

$$\|\tilde{r}^u + \tilde{u} - u_0\|_{\mathcal{X}}^2 + \|\tilde{r}^v + \tilde{v} - v_0\|_{\mathcal{Y}}^2 + 2\tilde{\varepsilon} \leq \sigma^2 \left(\|\tilde{u} - u_0\|_{\mathcal{X}}^2 + \|\tilde{v} - v_0\|_{\mathcal{Y}}^2 \right). \tag{125}$$

This section presents a scheme based on the Nesterov's accelerated variant of Section 3.1 for solving problem ($P2$) where $\widehat{\Psi}$ has the bilinear structure

$$\widehat{\Psi}(u, v) = f(u) + \langle Au, v \rangle + g_1(u) - g_2(v), \quad \forall (u, v) \in X \times Y \tag{126}$$

and conditions introduced in Section 2.2 hold.

We now make two remarks about problem ($P2$). First, finding the solution of the exact version of problem ($P2$), i.e., the one in which $\sigma = 0$, is equivalent to finding the unique saddle-point of

$$\min_{u \in X} \max_{v \in Y} \widehat{\Psi}(u, v) + \frac{1}{2}\|u - u_0\|^2 - \frac{1}{2}\|v - v_0\|^2 \tag{127}$$

where $\widehat{\Psi}$ is given by (126). More specifically, if $(\tilde{u}, \tilde{v})$ is the exact saddle-point of the above problem, then $(\tilde{u}, \tilde{v})$ and the quantities $(\tilde{r}^u, \tilde{r}^v) := (u_0 - \tilde{u}, v_0 - \tilde{v})$ and $\tilde{\varepsilon} := 0$ satisfy (124) and (125) with $\sigma = 0$. Second, although the above saddle-point problem has essentially the same structure as the one we are interested in solving, namely (6), its primal function (see (18)) has the key property that it is the composite sum of the easy convex nonsmooth function g_1 and a smooth convex function with Lipschitz continuous gradient. Hence, approximate solutions of (127) can be obtained by using a Nesterov's accelerated variant for composite convex optimization problems (e.g., the one in Subsection 3.1).

In view of the two observations above, it is reasonable to expect that approximate solutions of (127) yield solutions of problem ($P2$) (with $\sigma > 0$). Rather than tackling

the latter issue in an abstract setting, we instead propose a scheme based on the Nesterov's accelerated variant of Subsection 3.1 applied to (127) to obtain a solution of problem $(P2)$ and derive its corresponding iteration complexity.

We next discuss how the composite saddle-point problem (127) can be viewed as a composite convex optimization problem (38) satisfying conditions A.1-A.3. Clearly, (127) is a special case of (38) in which

$$\psi(u) := f(u) + \tilde{\phi}(u), \qquad g(u) := g_1(u) + \frac{1}{2}\|u - u_0\|_{\mathcal{X}}^2 \tag{128}$$

and

$$\tilde{\phi}(u) := \max_{v} \left\{ \phi(u, v) := \langle Au, v\rangle - g_2(v) - \frac{1}{2}\|v - v_0\|_{\mathcal{Y}}^2 \right\}. \tag{129}$$

It is apparent that the above function g satisfies condition A.1. The following result implies that the above ψ satisfies conditions A.2 and A.3. Its proof for the case in which Y is compact is well-known (see for example [40]). Since we are not assuming that the latter condition, we include for sake of completeness a simple proof for the more general version given below. Its statement uses the following notion of the induced norm of a linear operator $A : \mathcal{X} \to \mathcal{Y}$ defined as

$$\|A\| := \max_{x}\{\|Ax\|_{\mathcal{Y}} : \|x\|_{\mathcal{X}} \leq 1\}.$$

Proposition 5.2.1. *The following statements hold:*

(a) for every $u \in \mathcal{X}$, the maximization problem in (129) *has a unique optimal solution $v(u)$, i.e.,*

$$v(u) := \arg\max_{v} \langle Au, v\rangle - g_2(v) - \frac{1}{2}\|v - v_0\|_{\mathcal{Y}}^2; \tag{130}$$

(b) $\tilde{\phi}$ is convex, differentiable everywhere on $\mathcal{X}$, $\nabla\tilde{\phi}$ is $\|A\|^2$-Lipschitz continuous on $\mathcal{X}$ and

$$\nabla\tilde{\phi}(u) = A^* v(u) \quad \forall u \in \mathcal{X}; \tag{131}$$

(c) for every $u, \tilde{u} \in \mathcal{X}$,

$$l_{\tilde{\phi}}(u; \tilde{u}) = \phi(u, v(\tilde{u})). \tag{132}$$

Proof. (a) This statement follows immediately from the fact that the negative of the objective function of the max problem in (130) is proper, closed and strongly convex.

(b) Letting $\tilde{g}_2(v) := g_2(v) + \|v - v_0\|^2/2$ and using the definition of $\tilde{\phi}$ in (129), we easily see that

$$\tilde{\phi}(u) = \tilde{g}_2^*(Au) \quad \forall u \in \mathcal{X}. \tag{133}$$

Moreover, noting that $\tilde{g}_2$ is a proper closed strongly convex with modulus one, we conclude from Proposition 1.1.2 with $f = \tilde{g}_2$ that $\tilde{g}_2^*$ is differentiable everywhere on $\mathcal{Y}$ and $\nabla \tilde{g}_2^*$ is 1-Lipschitz continuous. The above two observations then easily imply that $\tilde{\phi}$ is convex, differentiable everywhere on $\mathcal{X}$ and $\nabla \tilde{\phi}$ is $\|A\|^2$-Lipschitz continuous on $\mathcal{X}$. Moreover, the optimality condition for (130) implies that $Au \in \partial \tilde{g}_2(v(u))$, and hence that $v(u) = \nabla \tilde{g}_2^*(Au)$ in view of Proposition 1.1.1(c). Now, (131) follows by differentiating (133) and using the latter conclusion.

(c) Using (131), and the definitions of $l_{\tilde{\phi}}(\cdot, \cdot)$, $\phi(\cdot, \cdot)$ and $v(u)$ in (11), (129) and (130), respectively, we easily see that

$$l_{\tilde{\phi}}(u; \tilde{u}) = \tilde{\phi}(\tilde{u}) + \langle \nabla \tilde{\phi}(\tilde{u}), u - \tilde{u} \rangle = \phi(\tilde{u}, v(\tilde{u})) + \langle A^* v(\tilde{u}), u - \tilde{u} \rangle = \phi(u, v(\tilde{u})).$$

□

In view of the above result, we conclude that the function ψ defined in (128) satisfies conditions A.2 and A.3 of Subsection 3.1 with $L = L_f + \|A\|^2$. We can then use Algorithm 1 to approximately solve (127), and hence $(P2)$ as will be shown later in this section.

We now state our accelerated scheme for solving problem (P2). It is essentially Algorithm 1 applied to (38) with $\widehat{\Psi}$ and g given by (128) and (129), respectively, endowed with two important refinements. The first one due to Nesterov (see (4.2) of

[40] or Corollary 3(c) of [60]) computes a dual iterate $\tilde{v}_k$ as in (134), which together with the primal iterate $\tilde{u}_k$, provides the first candidate pair $(\tilde{u}, \tilde{v}) = (\tilde{u}_k, \tilde{v}_k)$ for $(P2)$. The second one (see step 2 below) gives a recipe for computing the second candidate pair $(\tilde{r}^u, \tilde{r}^v) \in \mathcal{X} \times \mathcal{Y}$ and scalar $\tilde{\varepsilon} \geq 0$ which, together with the above pair $(\tilde{u}, \tilde{v})$, yield a candidate solution for $(P2)$.

[Algorithm 2] Accelerated method for problem $(P2)$**:**

Input: f, L_f, A, g_1 and g_2, $(u_0, v_0) \in \mathcal{X} \times \mathcal{Y}$ and $\sigma \in (0, 1)$.

0) set $L = L_f + \|A\|^2$, $\mu = 1$, $\Gamma_0 = 0$, $\tilde{u}_0 = w_0 = P_\Omega(u_0)$, $\tilde{v}_0 = 0$ and $k = 1$;

1) compute Γ_k, u_k and $v(u_k)$ as in (39), (40) and (130), respectively, $(\tilde{v}_k, w_k) \in Y \times X$ as

$$\tilde{v}_k := \frac{\Gamma_{k-1}}{\Gamma_k}\tilde{v}_{k-1} + \frac{\Gamma_k - \Gamma_{k-1}}{\Gamma_k} v(u_k), \tag{134}$$

$$w_k := \operatorname{argmin}_u l_{f,k}(u) + \langle A^*\tilde{v}_k, u\rangle + g_1(u) + \frac{c_k}{2}\|u - u_0\|_{\mathcal{X}}^2, \tag{135}$$

and $\tilde{u}_k$ as in (42), where

$$c_k := 1 + \frac{1}{\Gamma_k}, \qquad l_{f,k}(u) := \sum_{i=1}^{k} \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma_k} l_f(u; u_i); \tag{136}$$

2) set

$$\tilde{\varepsilon}_k = \frac{1}{2\Gamma_k}\|\tilde{u}_k - u_0\|_{\mathcal{X}}^2, \quad \tilde{r}_k^u := c_k(u_0 - w_k), \quad \tilde{r}_k^v := v_0 - v(\tilde{u}_k); \tag{137}$$

3) if $\|\tilde{r}_k^u + \tilde{u}_k - u_0\|_{\mathcal{X}}^2 + \|\tilde{r}_k^v + \tilde{v}_k - v_0\|_{\mathcal{Y}}^2 + 2\tilde{\varepsilon}_k \leq \sigma^2\|\tilde{u}_k - u_0\|_{\mathcal{X}}^2 + \sigma^2\|\tilde{v}_k - v_0\|_{\mathcal{Y}}^2$, then terminate and go to **Output**; otherwise, set $k \leftarrow k + 1$ and go to step 1.

Output: ouput $(\tilde{u}, \tilde{v}) = (\tilde{u}_k, \tilde{v}_k)$, $(\tilde{r}^u, \tilde{r}^v) = (\tilde{r}_k^u, \tilde{r}_k^v)$ and $\tilde{\varepsilon} = \tilde{\varepsilon}_k$.

The following simple result shows that step 1 of Algorithm 2 corresponds to an iteration of Algorithm 1 applied to (41) with ψ and g defined according to (128) and (129).

Lemma 5.2.2. *Let ψ and g be defined according to* (128) *and* (129). *Then, the following statements hold for every $k \geq 1$:*

(a) the function $l_{\psi,k}(u) - (l_{f,k}(u) + \langle A^*\tilde{v}_k, u\rangle)$ is constant;

(b) (135) *is equivalent to* (41).

Proof. (a) Relation (134) and the fact that $\Gamma_0 = 0$ imply that

$$\tilde{v}_k = \sum_{i=1}^{k} \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma_k} v(u_i). \tag{138}$$

Using the first identity in (128) and Proposition 5.2.1(b), we have that $\nabla\psi(u) = \nabla f(u) + A^*v(u)$, which together with definition (11) then imply that

$$l_\psi(u; u_i) = l_f(u; u_i) + [\tilde{\phi}(u_i) + \langle A^*v(u_i), u - u_i\rangle] \quad \forall i \geq 1.$$

Statement (a) now follows from the previous identity and relations (45), (136) and (138). (b) This statement immediately follows from (a), the definition of g in (128) and the definition of c_k in (136). □

Ignoring steps 2 and 3 of Algorithm 2 which are essentially computing $(\tilde{r}^u, \tilde{r}^v) = (\tilde{r}^u_k, \tilde{r}^v_k)$ and $\tilde{\varepsilon} = \tilde{\varepsilon}_k$ satisfying (124) and checking whether these entities, together with the primal-dual iterate $(\tilde{u}_k, \tilde{v}_k)$, satisfy (125), Lemma 5.2.2 immediately implies that Algorithm 2 is nothing more than Algorithm 0 applied to problem (38) with ψ and g given by (128).

The following technical result follows as a consequence of the latter observation and Proposition 3.1.1.

Lemma 5.2.3. *Consider the sequences $\{(\tilde{u}_k, \tilde{v}_k)\}$ generated by Algorithm 2 and define*

$$\tilde{\varepsilon}'_k := \frac{1}{2\Gamma_k}\|\tilde{u}_k - u_0\|^2_{\mathcal{X}} + l_{f,k}(\tilde{u}_k) - f(\tilde{u}_k), \tag{139}$$

$$\widehat{\Psi}_k(u, v) := l_{f,k}(u) + \langle Au, v\rangle + g_1(u) - g_2(v), \tag{140}$$

$$q_k(u, v) := \frac{c_k}{2}\|u - u_0\|^2_{\mathcal{X}} + \frac{1}{2}\|v - v_0\|^2_{\mathcal{Y}}, \tag{141}$$

where c_k and $l_{f,k}$ are defined in (136)*. Then,*

$$0 \in \partial_{\tilde{\varepsilon}'_k} \left[\widehat{\Psi}_k(\cdot, \tilde{v}_k) - \widehat{\Psi}_k(\tilde{u}_k, \cdot) + q_k(\cdot, \cdot) \right] (\tilde{u}_k, \tilde{v}_k), \tag{142}$$

Proof. Consider the functions ψ, g and ϕ defined in (128) and (129). It follows from (128), (129) and Proposition 3.1.1 that

$$\begin{aligned} f(\tilde{u}_k) + \phi(\tilde{u}_k, v) + g_1(\tilde{u}_k) + \frac{1}{2}\|\tilde{u}_k - u_0\|^2_{\mathcal{X}} &\leq (\psi + g)(\tilde{u}_k) \\ &\leq l_{\psi,k}(u) + g_1(u) + \frac{c_k}{2}\|u - u_0\|^2_{\mathcal{X}} \quad \forall (u, v) \in \mathcal{X} \times \mathcal{Y} \end{aligned}$$

where $l_{\psi,k}(\cdot)$ is defined in (45). Using the definitions of ψ and $\tilde{\phi}$ in (128) and (129), relation (11), definitions of $l_{\psi,k}(u)$ and $l_{f,k}(u)$ in (45) and (136), identities (132) and (138), and the fact $\phi(u, \cdot)$ is concave for any $u \in \mathcal{X}$, we conclude that

$$\begin{aligned} l_{\psi,k}(u) &= \sum_{i=1}^{k} \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma_k} \left(l_f(u; u_i) + l_{\tilde{\phi}}(u; u_i) \right) = l_{f,k}(u) + \sum_{i=1}^{k} \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma_k} \phi\left(u, v(u_i)\right) \\ &\leq l_{f,k}(u) + \phi\left(u, \sum_{i=1}^{k} \frac{\Gamma_i - \Gamma_{i-1}}{\Gamma_k} v(u_i)\right) = l_{f,k}(u) + \phi(u, \tilde{v}_k) \quad \forall u \in \mathcal{X}. \end{aligned}$$

Combining the above two relations and using the definition of ϕ, $\widehat{\Psi}_k$ and $\tilde{\varepsilon}'_k$ in (129), (140) and (139), respectively, we then conclude that

$$\begin{aligned} \widehat{\Psi}_k(\tilde{u}_k, v) - \frac{1}{2}\|v - v_0\|^2_{\mathcal{Y}} + \frac{c_k}{2}\|\tilde{u}_k - u_0\|^2_{\mathcal{X}} - \tilde{\varepsilon}'_k & \\ = l_{f,k}(\tilde{u}_k) + \phi(\tilde{u}_k, v) + g_1(\tilde{u}_k) + \frac{c_k}{2}\|\tilde{u}_k - u_0\|^2_{\mathcal{X}} - \tilde{\varepsilon}'_k & \\ \leq l_{f,k}(u) + \phi(u, \tilde{v}_k) + g_1(u) + \frac{c_k}{2}\|u - u_0\|^2_{\mathcal{X}} & \\ = \widehat{\Psi}_k(u, \tilde{v}_k) - \frac{1}{2}\|\tilde{v}_k - v_0\|^2_{\mathcal{Y}} + \frac{c_k}{2}\|u - u_0\|^2_{\mathcal{X}} \quad \forall (u, v) \in X \times Y. & \end{aligned} \tag{143}$$

Now, using the definition of the ε-differential in (12) and the definition of $q_k(\cdot, \cdot)$ in (141), the above inequality can be easily seen to be equivalent to (142).

□

The following result quantifies the quality of the entities $(\tilde{u}_k, \tilde{v}_k)$, $\tilde{\varepsilon}_k$ and $(\tilde{r}_k, \tilde{r}^v_k)$ generated at the k-th iteration of Algorithm 2 as a candidate solution for problem $(P2)$.

Lemma 5.2.4. *Consider the sequences* $\{(\tilde{u}_k, \tilde{v}_k)\}$, $\{\tilde{\varepsilon}_k\}$ *and* $\{(\tilde{r}_k, \tilde{r}_k^v)\}$ *generated by Algorithm 2. Then, for every* $k \geq 1$,

$$(\tilde{r}_k^u, \tilde{r}_k^v) \in \partial_{\tilde{\varepsilon}_k} \left[\widehat{\Psi}(\cdot, \tilde{v}_k) - \widehat{\Psi}(\tilde{u}_k, \cdot)\right](\tilde{u}_k, \tilde{v}_k), \tag{144}$$

$$\|\tilde{r}_k^u + \tilde{u}_k - u_0\|_{\mathcal{X}}^2 + \|\tilde{r}_k^v + \tilde{v}_k - v_0\|_{\mathcal{Y}}^2 + 2\tilde{\varepsilon}_k \leq \left(\frac{3}{\Gamma_k} + \frac{4}{\Gamma_k^2}\right)\|\tilde{u}_k - u_0\|_{\mathcal{X}}^2, \tag{145}$$

where $\widehat{\Psi}(\cdot)$ *is defined in* (126).

Proof. Equations (130) and (135) and the definitions of $\widehat{\Psi}_k$ and q_k in (140) and (141) imply that

$$(w_k, v(\tilde{u}_k)) = \arg\min_{(u,v)} \widehat{\Psi}_k(u, \tilde{v}_k) - \widehat{\Psi}_k(\tilde{u}_k, v) + q_k(u, v).$$

In view of the optimality condition of the above minimization problem, the definitions of $\tilde{r}_k^u$ and $\tilde{r}_k^v$ in (137), and the definition of $q_k(\cdot, \cdot)$ in (141), we then conclude that

$$(\tilde{r}_k^u, \tilde{r}_k^v) = -\nabla q_k(w_k, v(\tilde{u}_k)) \in \partial[\widehat{\Psi}_k(\cdot, \tilde{v}_k) - \widehat{\Psi}_k(\tilde{u}_k, \cdot)](w_k, v(\tilde{u}_k)).$$

Hence, by Proposition 1.1.1(b) we have

$$(\tilde{r}_k^u, \tilde{r}_k^v) = -\nabla q_k(w_k, v(\tilde{u}_k)) \in \partial_{\delta_k}[\widehat{\Psi}_k(\cdot, \tilde{v}_k) - \widehat{\Psi}_k(\tilde{u}_k, \cdot)](\tilde{u}_k, \tilde{v}_k). \tag{146}$$

where

$$\delta_k := -\left[\widehat{\Psi}_k(w_k, \tilde{v}_k) - \widehat{\Psi}_k(\tilde{u}_k, v(\tilde{u}_k))\right] - \langle -\nabla q_k(w_k, v(\tilde{u}_k)), (\tilde{u}_k, \tilde{v}_k) - (w_k, v(\tilde{u}_k))\rangle \geq 0.$$

On the other hand, in view of Lemma 5.2.3, inclusion (142) holds, or equivalently inequality (143) holds. The latter inequality with $(u, v) = (w_k, v(\tilde{u}_k))$, together with the definitions of $\tilde{\varepsilon}_k$ and $\tilde{\varepsilon}_k'$ in (137) and (139), then implies that

$$\begin{aligned}\tilde{\varepsilon}_k \geq \tilde{\varepsilon}_k' &\geq -\widehat{\Psi}_k(w_k, \tilde{v}_k) + \widehat{\Psi}_k(\tilde{u}_k, v(\tilde{u}_k)) + q_k(\tilde{u}_k, \tilde{v}_k) - q_k(w_k, v(\tilde{u}_k)) \\ &= \delta_k + \frac{c_k}{2}\|\tilde{u}_k - w_k\|_{\mathcal{X}}^2 + \frac{1}{2}\|\tilde{v}_k - v(\tilde{u}_k)\|_{\mathcal{Y}}^2,\end{aligned} \tag{147}$$

where the last equality comes from the definition of δ_k and the fact that the second order Taylor expansion of the quadratic function q_k at an arbitrary point agrees with

q_k itself. In view of (146) and (147), we then conclude that

$$(\tilde{r}_k^u, \tilde{r}_k^v) \in \partial_{\tilde{\varepsilon}_k'}[\widehat{\Psi}_k(\cdot, \tilde{v}_k) - \widehat{\Psi}_k(\tilde{u}_k, \cdot)](\tilde{u}_k, \tilde{v}_k).$$

Using the definition of ε-subdifferential in (12) and the definitions of $\tilde{\varepsilon}_k$, $\tilde{\varepsilon}_k'$, $\widehat{\Psi}$ and $\widehat{\Psi}_k$ in (137), (139), (126) and (140), respectively, and the fact that $\widehat{\Psi}_k(\cdot, \tilde{v}_k)$ is majorized by $\widehat{\Psi}(\cdot, \tilde{v}_k)$, it is now easy to see that the above inclusion implies (144).

Moreover, inequality (147), the definitions of $\tilde{r}_k^u$, $\tilde{r}_k^v$ and $\tilde{\varepsilon}_k$ in (137), and the fact $c_k = 1 + 1/\Gamma_k$ imply that

$$\begin{aligned}
&\|\tilde{r}_k^u + \tilde{u}_k - u_0\|_{\mathcal{X}}^2 + \|\tilde{r}_k^v + \tilde{v}_k - v_0\|_{\mathcal{Y}}^2 + 2\tilde{\varepsilon}_k \\
=&\|(u_0 - \tilde{u}_k)/\Gamma_k + c_k(\tilde{u}_k - w_k)\|_{\mathcal{X}}^2 + \|\tilde{v}_k - v(\tilde{u}_k)\|_{\mathcal{X}}^2 + 2\tilde{\varepsilon}_k \\
\leq&\frac{2}{\Gamma_k^2}\|\tilde{u}_k - u_0\|_{\mathcal{X}}^2 + 2c_k^2\|\tilde{u}_k - w_k\|_{\mathcal{X}}^2 + \|\tilde{v}_k - v(\tilde{u}_k)\|_{\mathcal{X}}^2 + 2\tilde{\varepsilon}_k \\
\leq&\frac{2}{\Gamma_k^2}\|\tilde{u}_k - u_0\|_{\mathcal{X}}^2 + (4c_k + 2)\tilde{\varepsilon}_k = \left(\frac{3}{\Gamma_k} + \frac{4}{\Gamma_k^2}\right)\|\tilde{u}_k - u_0\|_{\mathcal{X}}^2.
\end{aligned}$$

□

As an immediate consequence of Lemma 5.2.4, we can now derive the iteration-complexity for Algorithm 2 to solve problem $(P2)$.

Proposition 5.2.5. *Algorithm 2 terminates in at most*

$$\mathcal{O}\left(\left\lceil\sqrt{(L_f + \|A\|^2)\lceil\sigma^{-2}\rceil}\right\rceil\right) \tag{148}$$

iterations with an output which solves problem $(P2)$.

Proof. The inclusion (144) and the termination criterion in step 3 of Algorithm 2 show that the output of Algorithm 2 solves problem $(P2)$. To show the corollary, it suffices to show that Algorithm 2 finishes in at most

$$k_0 := \left\lceil 4\sqrt{(L_f + \|A\|^2)\lceil\sigma^{-2}\rceil}\right\rceil = \left\lceil 4\sqrt{L\lceil\sigma^{-2}\rceil}\right\rceil. \tag{149}$$

iterations, where the second equality is due to the definition of L in step 0 of Algorithm 2. Indeed, assume for contradiction that Algorithm 2 has not terminated at an

iteration $k > k_0$. The latter condition on k together with (44) and (149) can be easily seen to imply that

$$\Gamma_k > 4\max\{1, \sigma^{-2}\}.$$

Moreover, since Algorithm 2 has not terminated at the k-th iteration of Algorithm 2, it follows from the termination criterion on its step 3 and relation (145) that

$$\sigma^2\|\tilde{u}_k - u_0\|_{\mathcal{X}}^2 < \|\tilde{r}_k^u + \tilde{u}_k - u_0\|_{\mathcal{X}}^2 + \|\tilde{r}_k^v + \tilde{v}_k - v_0\|_{\mathcal{Y}}^2 + 2\tilde{\varepsilon}_k \leq \left(\frac{3}{\Gamma_k} + \frac{4}{\Gamma_k^2}\right)\|\tilde{u}_k - u_0\|_{\mathcal{X}}^2.$$

Hence, it follows from the above two conclusions that

$$\sigma^2 < \left(\frac{3}{\Gamma_k} + \frac{4}{\Gamma_k^2}\right) < \frac{4}{\Gamma_k} < \sigma^2.$$

□

5.3 Accelerated algorithm Acc-SP-HPE for solving a special class of composite saddle-point problem

This section presents a special instance of the SP-HPE framework introduced in Section 5.1, which we refer to as the Acc-SP-HPE method, for solving the class of composite saddle-point problem (6), or equivalently, the saddle-point problem $SP(\widehat{\Psi}; X, Y)$ with $\widehat{\Psi}$ defined in (126). Each (outer) iteration of the Acc-SP-HPE method, which is essentially a special iteration of the SP-HPE framework, invokes Algorithm 2 to obtain a solution of the inexact prox subproblem (119)-(120). A complexity bound on the total number of iterations performed by Algorithm 2 (called the inner iterations) performed by the Acc-SP-HPE method to find a (ρ, ε)-saddle-point is derived in section. Moreover, an inner-iteration complexity for the Acc-SP-HPE method to find an ε-saddle-point is also derived for the case when the feasible set $X \times Y$ is bounded.

We assume in this section that the solution set of the composite saddle-point problem (6) is nonempty, and assumptions C.1-C.3 are satisfied. Initially, we do not assume boundedness of the feasible set $X \times Y$. The case where $X \times Y$ is assumed to be bounded will be discussed in Subsection 5.3.1.

Recall that in Section 5.2 we have motivated the introduction of problem $(P2)$ as a special case of the inexact prox subproblem (119)-(120) in which $\lambda = 1$. The following result shows in fact that problem $(P2)$ is as general as subproblem (119)-(120) for any value of $\lambda > 0$.

Proposition 5.3.1. *Let $\lambda > 0$ and $\widehat{\Psi}$ be a closed convex-concave function and consider the k-th iteration of the SP-HPE framework. If $(\tilde{u}, \tilde{v}) \in \mathcal{X} \times \mathcal{Y}$, $(\tilde{r}^u, \tilde{r}^v) \in \mathcal{X} \times \mathcal{Y}$ and $\tilde{\varepsilon} \geq 0$ solve problem $(P2)$ with input $\widehat{\Psi} = \lambda\widehat{\Psi}$, $(u_0, v_0) = (x_{k-1}, y_{k-1})$ and $\sigma > 0$, then*

$$(\tilde{x}_k, \tilde{y}_k) := (\tilde{u}, \tilde{v}), \quad (\tilde{r}_k^x, \tilde{r}_k^y) := \frac{1}{\lambda}(\tilde{r}^u, \tilde{r}^v), \quad \varepsilon_k := \frac{1}{\lambda}\tilde{\varepsilon}$$

satisfy the conditions (119) *and* (120) *of step 1 of the SP-HPE framework.*

Proof. The conclusion follows immediately from the identity

$$\lambda\partial_\varepsilon \left[\widehat{\Psi}(\cdot, \tilde{v}) - \widehat{\Psi}(\tilde{u}, \cdot)\right](\tilde{u}, \tilde{v}) = \partial_{\lambda\varepsilon} \left[\lambda\widehat{\Psi}(\cdot, \tilde{v}) - \lambda\widehat{\Psi}(\tilde{u}, \cdot)\right](\tilde{u}, \tilde{v})$$

which holds for every $\varepsilon \geq 0$, $\lambda \geq 0$ and $(\tilde{u}, \tilde{v}) \in \mathcal{X} \times \mathcal{Y}$. □

In view of the above result, we can use Algorithm 2 to solve the inexact prox subproblem (119)-(120). This is the key idea behind the following special case of the SP-HPE framework, referred to as the Acc-SP-HPE method, for solving the saddle-point problem (6).

[Acc-SP-HPE] Accelerated SP-HPE Method for solving problem (6)**:**

0) Let $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$, $\lambda > 0$ and $0 < \sigma < 1$ be given and set $k = 1$;

1) invoke Algorithm 2 with input

$$f = \lambda f, \quad A = \lambda A, \quad g_1 = \lambda g_1, \quad g_2 = \lambda g_2, \quad (u_0, v_0) = (x_{k-1}, y_{k-1}), \quad L_f = \lambda L_f,$$

and set

$$(\tilde{x}_k, \tilde{y}_k) := (\tilde{u}, \tilde{v}), \quad \tilde{r}_k = (\tilde{r}_k^x, \tilde{r}_k^y) := \frac{1}{\lambda}(\tilde{r}^u, \tilde{r}^v), \quad \varepsilon_k := \frac{1}{\lambda}\tilde{\varepsilon}$$

where $(\tilde{u}, \tilde{v})$, $(\tilde{r}^u, \tilde{r}^v)$ and $\tilde{\varepsilon}$ are the output generated by Algorithm 2;

2) set $x_k = x_{k-1} - \lambda \tilde{r}_k^x, \quad y_k = y_{k-1} - \lambda \tilde{r}_k^y$, set $k \leftarrow k+1$, and go to step 1.

end

Proposition 5.3.2. *Acc-SP-HPE method is a special case of the SP-HPE framework for solving the composite saddle-point problem* (6).

Proof. In view of Proposition 5.3.1, the sequences $\{(\tilde{x}_k, \tilde{y}_k)\}$, $\{(\tilde{r}_k^x, \tilde{r}_k^y)\}$ and $\{\varepsilon_k\}$ generated by the Acc-SP-HPE method satisfy the conditions (119) and (120) of step 1 of the SP-HPE framework. Therefore, Acc-SP-HPE method is clearly a special case of the SP-HPE framework. □

It follows as a consequence of Proposition 5.3.2 that the pointwise and ergodic (outer) convergence rate bounds for the Acc-SP-HPE method are as described in statements (a) and (b) of Theorem 5.1.2, respectively.

Theorem 5.3.3. *Assume that conditions C.1-C.3 hold,* $\max\{\sigma^{-1}, (1-\sigma)^{-1}\} = \mathcal{O}(1)$ *and the (convex) set of saddle-points of* (6) *is non-empty, and let* d_0 *denote the distance of the initial iterate* (x_0, y_0) *of the Acc-SP-HPE method with respect to this set. Consider the sequences* $\{(\tilde{x}_k, \tilde{y}_k)\}$, $\{(\tilde{r}_k^x, \tilde{r}_k^y)\}$ *and* $\{\varepsilon_k\}$ *generated by the Acc-SP-HPE method and the ergodic sequences* $\{(\tilde{x}_k^a, \tilde{y}_k^a)\}$, $\{\tilde{r}_k^a\}$ *and* $\{\varepsilon_k^a\}$ *defined in Theorem 5.1.2. Then, the following statements hold:*

(a) for every pair of positive scalars (ρ, ε)*, there exists*

$$k_0 = \mathcal{O}\left(\max\left\{1, \frac{d_0}{\lambda\rho}, \frac{d_0^2}{\lambda\varepsilon}\right\}\right)$$

such that for every $k \geq k_0$*, the triple* $((\tilde{x}_k^a, \tilde{y}_k^a), \tilde{r}_k^a, \varepsilon_k^a)$ *is a* (ρ, ε)*-saddle-point of* (6)*;*

(b) each iteration of the Acc-SP-HPE method performs at most

$$\mathcal{O}\left(\left\lceil \sqrt{\lambda L_f + \lambda^2 \|A\|^2} \right\rceil\right)$$

inner iterations (and hence resolvent evaluations of ∂g_1 *and* ∂g_2*).*

As a consequence, the Acc-SP-HPE method finds a (ρ, ε)-saddle-point of (6) *by performing no more than*

$$\mathcal{O}\left(\left\lceil \sqrt{(\lambda L_f + \lambda^2 \|A\|^2)} \right\rceil \max\left\{1, \frac{d_0}{\lambda\rho}, \frac{d_0^2}{\lambda\varepsilon}\right\}\right) \tag{150}$$

inner iterations (and hence resolvent evaluations of ∂g_1 and ∂g_2).

Proof. Since by Proposition 5.3.2 the Acc-SP-HPE method is a special instance of the SP-HPE framework, (a) follows immediately from Theorem 5.1.2(b). Statement (b) follows immediately from Proposition 5.2.5 with $L_f = \lambda L_f$ and $A = \lambda A$, and the fact that each iteration of Algorithm 2 performs one resolvent evaluation of ∂g_1 and two resolvent evaluations of ∂g_2. The last assertion of the theorem follows immediately from (a) and (b). □

We now make some remarks about possible values of λ which minimize the complexity bound (150) (up to an additive and multiplicative $\mathcal{O}(1)$ constant). Noting that (150) is equivalent to

$$\mathcal{O}\left(\max\left\{\frac{1}{\lambda}, \sqrt{\frac{L_f}{\lambda}}, \|A\|\right\} \max\left\{\lambda, \frac{d_0}{\rho}, \frac{d_0^2}{\varepsilon}\right\}\right)$$

then it is straightforward to see that the following claims hold depending on whether the condition

$$\lambda_1 := \max\left\{\frac{L_f}{\|A\|^2}, \frac{1}{\|A\|}\right\} \leq \max\left\{\frac{d_0}{\rho}, \frac{d_0^2}{\varepsilon}\right\} =: \lambda_2 \tag{151}$$

holds (case 1) or not (case 2):

1) if (151) holds then any $\lambda \in [\lambda_1, \lambda_2]$ minimizes (150) with minimum value equal to

$$\mathcal{O}\left(\|A\| \max\left\{\frac{d_0}{\rho}, \frac{d_0^2}{\varepsilon}\right\}\right);$$

2) otherwise, if $\lambda_1 > \lambda_2$, then $\lambda = \lambda_2$ minimizes (150) with minimum value equal to

$$\mathcal{O}\left(1 + \sqrt{L_f} \max\left\{\sqrt{\frac{d_0}{\rho}}, \frac{d_0}{\sqrt{\varepsilon}}\right\}\right).$$

Ideally, one should choose λ according to the above discussion in order to minimize the total number of resolvent evaluations of ∂g_1 and ∂g_2. But, since d_0 is usually not known a priori, we can not compute λ_2, and as a result choose $\lambda = \lambda_2$ as proposed in case 2 above. Note however that we can always choose $\lambda = \lambda_1$ since the latter is easily computable. Clearly, this choice is optimal when case 1 holds and, even though is not optimal when case 2 holds, we believe it might be a good practical choice in both cases due to the fact that case 2 is quite unlikely.

5.3.1 Scaling of Acc-SP-HPE method for bounded composite saddle-point problem

In this subsection, we consider the special case of problem (6) where the feasible set $X \times Y$ is bounded and derive a complexity bound on the number of inner iterations performed by the Acc-SP-HPE method to find an ε-saddle-point.

Corollary 5.3.4. *Suppose that the assumptions of Theorem 5.3.3 hold,* $(x_0, y_0) \in X \times Y$ *and the diameter* D *of the set* $X \times Y$ *defined in* (9) *is finite. Then, for any* $\varepsilon > 0$*, Acc-SP-HPE method finds an* ε*-saddle-point of* (6) *by performing no more than*

$$\mathcal{O}\left(\left\lceil \sqrt{(\lambda L_f + \lambda^2 \|A\|^2)} \right\rceil \max\left\{1, \frac{d_0 D}{\lambda \varepsilon}\right\}\right) \leq \mathcal{O}\left(\left\lceil \sqrt{(\lambda L_f + \lambda^2 \|A\|^2)} \right\rceil \max\left\{1, \frac{D^2}{\lambda \varepsilon}\right\}\right) \tag{152}$$

resolvent evaluations of ∂g_1 *and* ∂g_2*.*

Proof. Under the assumption that D is finite, it is straightforward to see from Definition 2.1.1 and the definition of subdifferential that an $(\varepsilon/2D, \varepsilon/2)$-saddle-point is always an ε-saddle-point. The first bound in (152) now follows immediately from the fact that $d_0 \leq D$ in view of the assumption that $(x_0, y_0) \in X \times Y$, and from the bound (150) in Theorem 5.3.3 with $(\rho, \varepsilon) = (\varepsilon/(2D), \varepsilon/2)$. Clearly, $d_0 \leq D$ also implies the second bound in (152). □

We now make a few comments about choosing λ so as to minimize the right hand side of (152) (up to an additive and multiplicative $\mathcal{O}(1)$ constant). Similar to the discussion in the previous subsection, if

$$\widehat{\lambda}_1 := \max\left\{\frac{L_f}{\|A\|^2}, \frac{1}{\|A\|}\right\} \leq \frac{D^2}{\varepsilon} =: \widehat{\lambda}_2 \tag{153}$$

holds, then any $\lambda \in [\widehat{\lambda}_1, \widehat{\lambda}_2]$ minimizes the right hand side of (152) with minimum value equal to $\mathcal{O}(1 + D^2\|A\|/\varepsilon)$. Otherwise, if $\widehat{\lambda}_1 > \widehat{\lambda}_2$, then $\lambda = \widehat{\lambda}_2$ minimizes the right hand side of (152) with minimum value equal to $\mathcal{O}(1 + D\sqrt{L_f/\varepsilon})$. Observe that regardless of which case holds, the right hand side of (152) assume its minimum value

$$\mathcal{O}\left(1 + D^2\frac{\|A\|}{\varepsilon} + D\sqrt{\frac{L_f}{\varepsilon}}\right) \tag{154}$$

when $\lambda = \min\left\{\widehat{\lambda}_1, \widehat{\lambda}_2\right\}$.

Clearly, letting D_X and D_Y denote the diameter of X and Y, we have $D = (D_X^2 + D_Y^2)^{1/2}$. Hence, we have $D_X \leq D$ and $D_X D_Y \leq D^2/2$, and it is clearly possible that $D_X << D$ and/or $D_X D_Y << D^2/2$. The rest of this subsection shows that the Acc-SP-HPE method applied to problem (6) with $\mathcal{X}$ and $\mathcal{Y}$ endowed with suitable scaled inner products has a resolvent complexity similar to (154) but with D^2 in the first term replaced by $D_X D_Y$ and D in the second term replaced by D_X.

To achieve the above goal, we endow $\mathcal{X}$ and $\mathcal{Y}$ with new inner products

$$\langle\cdot,\cdot\rangle_{\mathcal{X},\theta} := \theta\langle\cdot,\cdot\rangle, \quad \langle\cdot,\cdot\rangle_{\mathcal{Y},\theta} := \theta^{-1}\langle\cdot,\cdot\rangle, \tag{155}$$

respectively, and the associated norms then become

$$\|\cdot\|_{\mathcal{X},\theta} := \theta^{1/2}\|\cdot\|_{\mathcal{X}}, \quad \|\cdot\|_{\mathcal{Y},\theta} := \theta^{-1/2}\|\cdot\|_{\mathcal{Y}}$$

and problem (6) becomes

$$\min_{x\in X}\max_{y\in Y}\widehat{\Psi}(x,y) = f(x) + \langle A_\theta x, y\rangle_{\mathcal{Y},\theta} + g_1(x) - g_2(y), \tag{156}$$

where $A_\theta := \theta A$. Moreover, $\|A_\theta\|_\theta = \|A\|$ where $\|C\|_\theta := \max_x\{\|Cx\|_{\mathcal{Y},\theta} : \|x\|_{\mathcal{X},\theta} \leq 1\}$ and the gradient of f with respect to $\langle\cdot,\cdot\rangle_{\mathcal{X},\theta}$ is $L_{f,\theta}$-Lipschitz continuous on Ω where $L_{f,\theta} = \theta^{-1}L_f$. Also, the diameter of the feasible set $X \times Y$ with the product space $\mathcal{X} \times \mathcal{Y}$ endowed with the Cartesian inner product $\langle\cdot,\cdot\rangle_{\mathcal{X},\theta} + \langle\cdot,\cdot\rangle_{\mathcal{Y},\theta}$ is

$$D_\theta^2 := \theta D_X^2 + \theta^{-1} D_Y^2.$$

Using the above observations, we immediately see that the Acc-SP-HPE method applied to problem (6) where θ and λ are chosen as

$$\theta = \frac{D_Y}{D_X}, \qquad \lambda = \min\left\{\max\left\{\frac{L_f D_X}{\|A\|^2 D_Y}, \frac{1}{\|A\|}\right\}, \frac{2D_X D_Y}{\varepsilon}\right\},$$

and $\mathcal{X}$ and $\mathcal{Y}$ are endowed with the inner products (155), computes an ε-saddle-point of (6) by performing no more than

$$\mathcal{O}\left(1 + \frac{\|A\|}{\varepsilon} D_X D_Y + \sqrt{\frac{L_f}{\varepsilon}} D_X\right) \tag{157}$$

resolvent evaluations of ∂g_1 and ∂g_2.

It is worth noting that the above complexity is the same as the complexity of Nesterov's smoothing method (see (4.4) in [40]).

CHAPTER VI

NUMERICAL EXPERIMENTS

In this chapter, we conduct experiments to evaluate the performance of Acc-BD and Acc-SP-HPE algorithms on a collection of saddle-point and/or convex optimization problems.

The numerical performance of the two new methods is compared with several previous methods including T-BD (see Section 4.2) and Korpelevich's extragradient method (Korp) [25]. Since the latter two methods, as well as the two methods studied in this paper, are special cases of the HPE framework first proposed in [52], we have used in their implementation an adaptive stepsize strategy (see [34]) which takes the largest extragradient stepsize satisfying the HPE relative error criteria. It is worth noting that this stepsize can be obtained by solving an easy quadratic equation. All of these four methods can be further accelerated by using a dynamic scaling technique discussed in [38, 34] to properly balance the magnitude of the primal and dual residuals. However, we have not included this technique in our implementation of these four methods (except in Section 6.1) since its implementation is complex and time-consuming. The true values of the Lipschitz constants L_{xx}, L_{yy} and L_{xy}, all computed with respect to Euclidean norm, are used for these four methods. We also note our implementation of Acc-BD incorporates the safeguard that the subproblems (73) and/or (74) are solved (exactly) using the recipe of Proposition 4.2.1 whenever $L_{xx} \leq L_{xy}$ and/or $L_{yy} \leq L_{xy}$, respectively.

We have also compared the four methods above with two other well-known methods, namely: Nemirovski's prox-method (referred to Nemi-prox) [39, 60] and Nesterov's smooth approximation scheme [40] (referred to Nest-app) where the smooth

approximation is solved by a variant of Nesterov's optimal method due to Tseng, namely Algorithm 3 of [60] based on the update formula (18) there. We observe that Nemi-prox is an extension of Korpelevich's extragradient method which is based on a general distance generating function (e.g., the entropy function $\sum_i x_i \log x_i$) instead of the standard one, namely $\|\cdot\|^2/2$, used by Korpelevich's method. Our implementation of Nemi-prox uses the L_1-norm on $\mathcal{X} \times \mathcal{Y}$ and the entropy distance-generating function (see pages 15-16 of [39]). Nest-app approximates the non-smooth max component of the objective function by adding a small positive multiple of the entropy function to the max objective function and then applies the aforementioned Tseng's variant based on the entropy function to solve the resulting smooth approximation. The latter method endows both $\mathcal{X}$ and $\mathcal{Y}$ with the L_1-norm (see pages 149-150 of [40]). To improve the performance of these two methods, their implementation follows the recipe given in [60], i.e., the initial value of the Lipschitz constant is set to a fraction (1/8 was used in [60] and also in our experiments) of its true value and is increased by a factor of 2 whenever a certain convergence criterion (see equations (23) and (45) of [60]) is not satisfied.

We now make some observations about the way our computational results are presented. First, for problems with bounded feasible sets $X \times Y$ such as the ones considered in Sections 6.1 and 6.2, we have used the duality gap criterion of finding $(x, y) \in X \times Y$ such that $\text{gap}(x, y) \leq \varepsilon$ (see (22)) to terminate all methods due to the fact that Nest-app and/or Nemi-prox have been originally designed for the latter termination criterion. Second, we have excluded Nest-app from the comparison in Section 6.2 due to the fact it has to solve the perturbed max subproblem exactly in order to compute the gradient of the smooth approximation of the original objective function and the fact that this subproblem is expensive for the saddle-point problem considered in this section. Third, we have excluded both Nest-app and Nemi-prox from the comparison in Section 6.3 since the methods considered there are terminated

based on the notion of approximate saddle-point of Definition 2.1.1. The reason for changing the termination criterion on this section is due to the fact that its saddle-point problem has unbounded feasible set and the fact that none of the six methods compared in this paper have been shown to converge based on the (stronger) duality gap criterion in this situation.

Finally, all the computational results were obtained in MATLAB R2013a on a quad-core Linux machine with 8GB memory.

6.1 Vector-matrix saddle-point problem

This subsection compares Acc-BD with T-BD, Korp, Nemi-prox and Nest-app for solving a collection of instances of the minimization problem

$$\min_{x\in\Delta_m} \frac{1}{2}\|Cx-b\|^2 + \theta_{max}(A(x)), \tag{158}$$

where $C \in \Re^{m\times m}$, $b \in \Re^m$, $A_1, \cdots, A_m \in \mathcal{S}^n$ and $A(x) = \sum_{i=1}^m x_i A_i \in \mathcal{S}^{n\times n}$. It is easy to verify that the above problem is equivalent to the following vector-matrix saddle-point problem:

$$\min_{x\in\Delta_m} \max_{y\in\Omega} \Psi_1(x,y) = \frac{1}{2}\|Cx-b\|^2 + \langle A(x), y\rangle \tag{159}$$

where $\Omega = \{y \in \mathcal{S}^n : tr(y) = 1, y \succeq 0\}$.

Hence, we can apply the above methods on the saddle-point problem (159). In the numerical experiment, the matrices $A_1, \cdots, A_m$ and C are randomly generated such that each entry is generated independently and uniformly in the interval $[-1, 1]$ and $A_1, \cdots, A_m$ are then symmetrized. All methods are terminated whenever the duality gap at a candidate solution $(\tilde{x}, \tilde{y})$ is less than a given tolerance ϵ, i.e.,

$$\frac{1}{2}\|C\tilde{x}-b\|^2 + \theta_{max}(A(\tilde{x})) - \min_{x\in\Delta_m}\left\{\frac{1}{2}\|Cx-b\|^2 + \langle A(x), \tilde{y}\rangle\right\} \le \epsilon. \tag{160}$$

Both the current pointwise iterate $(\tilde{x}_k, \tilde{y}_k)$ and the current ergodic iterate $(\tilde{x}^a_k, \tilde{y}^a_k)$ defined in (82) are used to check the stopping criterion (160) for the methods Acc-BD,

T-BD, Korp and Nemi-prox. As described in Theorem 3 of [40] (see also Corollary 3 of [60]), the usual dual sequence generated by Nest-app is obtained by taking a weighted average of a sequence of dual maximizers for the perturbed max subproblems. In our experiment, we evaluate the max term of (160) at the current (usual) primal iterate and the min term of (160) at both the current weighted average dual iterate and the current dual maximizer, and choose the largest of the two values in order to obtain the smallest value for (160).

Table 1 reports the CPU time and the number of eigen-decompositions (in order to evaluate the resolvent of $\partial \mathcal{I}_\Omega$) for each method. The CPU times reported in this table do not include the time spent to evaluate the left hand side of stopping criterion (160), which is checked every 5 iterations. Due to space limitation, Table 1 does not specify the number of iterations performed by each method. We note however that the number of iterations performed by T-BD, Korp, Nemi-prox and Nest-app can be obtained by dividing the corresponding number of eigen-decompositions by $1, 2, 2$ and 2, respectively. Also, the number of outer (HPE) iterations performed by Acc-BD is equal to the number of eigen-decompositions due to the fact that $L_{yy} = 0$ for the saddle-point considered in this subsection and the fact that the safeguard used in our implementation ensures that the proximal subproblem in the y-variable is solved by means of a single resolvent evaluation of $\partial \mathcal{I}_\Omega$.

Observe from the results reported on Table 1 that the four HPE methods performed better than both Nemi-prox and Nest-app on this collection of saddle-point problems. We believe that this might be due to the fact that the implementation of these methods incorporate both the adaptive stepsize and scaling strategies mentioned above (see [38] and [34] for details on these strategies). Also, Acc-BD was by far the fastest among the six methods on this collection of saddle-point instances.

Table 1: Computational results for the methods Acc-BD, T-BD, Korp, Nemi-prox and Nest-app on vector-matrix saddle-point problems (159) with different sizes. All methods are terminated using criterion (160) with $\epsilon = 10^{-4}$ and 10^{-5}. CPU time in seconds and the number of eigen-decompositions are reported for each method.

Tol.	Problem	Acc-BD	T-BD	Korp	Nemi-prox	Nest-app
ε	m/n/$\frac{L_{xx}}{L_{xy}}$	time /#eigen	time /#eigen	time /#eigen	time /#eigen	time /#eigen
10^{-4}	100/50/4.73	**0.34**/ 50	0.62 / 200	0.74 / 400	4.09 / 2730	68.04 / 33310
	100/100/2.66	**2.25** / 185	3.66 / 360	5.83 / 910	42.74 /6720	270.63 / 34650
	100/200/1.55	**9.12** / 210	18.48 / 530	11.88 / 520	77.06 / 3220	924.61 / 31200
	200/50/9.30	**0.83** /150	1.66 / 350	2.45 / 860	5.05 / 2060	101.02 / 29030
	200/100/5.32	**2.12** / 105	6.16 / 410	9.98 / 1080	25.88 / 3080	388.10 / 30520
	200/200/2.72	**10.85** / 160	19.60 /360	28.22 / 830	216.60 / 6620	1877.7 / 32150
	500/50/19.88	**1.38** /80	2.94 / 315	3.21 / 600	16.78 / 3600	147.75 / 19680
	500/100/12.16	**5.31** / 130	22.49 / 790	17.06 / 1090	199.88 / 12870	675.36 / 20970
	500/200/6.98	**35.81** / 265	141.43 / 1315	113.68 / 1890	489.88 / 8230	3346.9 / 21500
10^{-5}	100/50/4.73	**0.50**/ 150	0.78 / 230	1.01 / 540	11.77 / 7240	N/A / >200000
	100/100/2.66	**4.51** / 345	5.14 / 515	8.40 / 1340	89.70 / 14520	N/A / >200000
	100/200/1.55	**19.38** / 420	31.11 /915	18.06 / 730	142.10 / 6570	N/A / >200000
	200/50/9.30	**1.51** / 305	2.15 / 500	4.37 / 1530	9.75 / 3760	N/A / >200000
	200/100/5.32	**4.28** / 225	8.73 / 605	14.35 / 1550	56.70 / 6290	N/A / >200000
	200/200/2.72	**15.87** / 220	27.19 / 500	45.64 / 1310	489.90 / 15090	N/A/ >200000
	500/50/19.88	**1.93** /130	4.16 / 440	4.47 /840	28.52/ 5870	N/A / >200000
	500/100/12.16	**6.06** / 150	26.47 / 925	22.41/ 1420	331.80 / 21430	N/A / >200000
	500/200/6.98	**66.12** / 445	185.11 / 1650	147.85 / 2500	1294.2 / 21260	N/A / >200000

6.2 Quadratic game problem

This subsection compares Acc-BD with T-BD, Korp and Nemi-prox for solving a collection of instances of the quadratic game problem

$$\min_{x\in\Delta_m}\max_{y\in\Delta_n}\Psi(x,y)=\frac{1}{2}\|Bx\|^2+x^\top Ay-\frac{1}{2}\|Cy\|^2 \tag{161}$$

where $A\in\Re^{m\times n}$, $B\in\Re^{m\times m}$ and $C\in\Re^{n\times n}$.

For this comparison, the matrices A, B and C are randomly generated such that each entry is nonzero with probability p and each nonzero entry is generated independently and uniformly in the interval $[0,1]$. The above five methods are terminated whenever the duality gap at the candidate solution $(\tilde{x},\tilde{y})$ is less than a given tolerance ϵ, i.e.,

$$\max_{y\in\Delta_n}\left\{\frac{1}{2}\|B\tilde{x}\|^2+\tilde{x}^\top Ay-\frac{1}{2}\|Cy\|^2\right\}-\min_{x\in\Delta_m}\left\{\frac{1}{2}\|Bx\|^2+x^\top A\tilde{y}-\frac{1}{2}\|C\tilde{y}\|^2\right\}\le\epsilon. \tag{162}$$

Both the iterate sequence $\{(\tilde{x}_k,\tilde{y}_k)\}$ and the ergodic sequence $\{(\tilde{x}^a_k,\tilde{y}^a_k)\}$ are used to check the stopping criterion (162) for the five methods considered in this section.

Table 2 reports the CPU time and the number of gradient evaluations (i.e., evaluations of $\nabla_x\Psi(\cdot,\cdot)$ and $\nabla_y\Psi(\cdot,\cdot)$, each counted separately) for each method. This table

also reports the number of outer (HPE) iterations for the Acc-BD method. The CPU times reported in this table do not include the time spent to evaluate the left hand side of stopping criterion (162), which is checked every outer iteration for Acc-BD and every five iterations for the other four methods. Due to space limitation, Table 2 does not specify the number of iterations performed by the methods T-BD, Korp and Nemi-prox . We note however that the number of iterations performed by these three methods can be obtained by dividing the corresponding number of gradient evaluations by 4.

Table 2 shows that T-BD had almost the same numerical performance as Korp on this collection of quadratic game instances and they are outperformed by Nemi-prox on several instances of this collection. Acc-BD was by far the fastest among the four methods on all instances of this collection. The results also confirm our conclusion in the paragraph following Corollary 4.4.2 that the performance of Acc-BD improves as the ratio $\max\{L_{xx}, L_{yy}\}/L_{xy}$ increases.

Table 2: Computational results for the methods Acc-BD, A-SP-HPE, T-BD and Korp on two-player quadratic games with different sizes and sparsities. All methods are terminated using criterion (162) with $\epsilon = 10^{-3}$ and 10^{-6}. CPU time in seconds and number of gradient evaluations are reported for each method.

Tol.	Problem size	Lip. ratio		Acc-BD		T-BD		Korp		Nemi-prox	
ε	m/n/p	$\frac{L_{xx}}{L_{xy}}$	$\frac{L_{yy}}{L_{xy}}$	time	#grad./iter.	time	#grad.	time	#grad.	time	#grad.
10^{-3}	1000/1000/0.1	48.11	48.03	**0.37**	276/7	0.86	700	0.94	720	1.68	1220
	1000/1000/0.2	91.11	91.46	**0.74**	378/8	2.92	1520	3.05	1540	1.72	780
	1000/2000/0.1	34.37	135.67	**0.85**	347/7	4.85	2080	4.98	2080	4.76	1880
	1000/2000/0.2	64.61	257.13	**2.22**	569/9	20.22	5120	20.53	5120	5.93	1420
	2000/1000/0.1	135.28	34.18	**0.77**	307/7	4.93	2160	5.20	2180	4.78	1920
	2000/1000/0.2	256.76	64.77	**2.00**	508/8	19.23	4900	19.55	4900	5.80	1440
	2000/2000/0.1	95.65	96.04	**0.93**	286/6	4.49	1220	4.57	1240	4.64	1220
	2000/2000/0.2	181.91	181.75	**2.25**	406/6	15.12	2480	15.64	2500	4.98	780
10^{-6}	1000/1000/0.1	48.11	48.03	**0.91**	802/32	2.47	2120	2.69	2140	14.24	10840
	1000/1000/0.2	91.11	91.46	**2.07**	1058/28	7.64	4000	8.15	4020	12.49	6060
	1000/2000/0.1	34.37	135.67	**2.86**	1188/38	17.87	7780	18.67	7780	36.25	14740
	1000/2000/0.2	64.61	257.13	**5.52**	1400/30	56.91	14540	58.89	14540	42.69	10100
	2000/1000/0.1	135.28	34.18	**2.07**	844/24	17.07	7480	17.90	3740	37.28	15120
	2000/1000/0.2	256.76	64.77	**5.10**	1256/26	49.00	15300	60.81	15300	39.65	9700
	2000/2000/0.1	95.65	96.04	**2.80**	790/20	13.34	3700	14.11	3720	42.72	11020
	2000/2000/0.2	181.91	181.75	**5.85**	1029/19	41.73	6760	42.57	6800	37.56	5880

6.3 A regularized least-square problem

This subsection examines the performance of methods Acc-BD, A-SP-HPE, T-BD and Korp for solving a collection of instances of the following regularized least-square

problem

$$\min_{X \in \Re^{k \times n}} \frac{1}{2}\|AX - B\|_F^2 + \beta\|X\|_1 + \gamma\|X\|_*, \tag{163}$$

where the matrices $A \in \Re^{m \times k}$, $B \in \Re^{m \times n}$ and the regularization parameters $\beta > 0$ and $\gamma > 0$ are given. Note that the purpose of the regularization term $\beta\|X\|_1 + \gamma\|X\|_*$ in (163) is to simultaneously induce sparsity and low-rankness on X. Clearly, problem (163) is a special instance of the class of optimization problem (28) where $g_1(X) = \beta\|X\|_1$ and $g_2^*(X) = \gamma\|X\|_*$. Hence we can apply the above four methods to solve problem (163).

In the numerical experiment, the matrices A and B are generated as sparse matrices with 1% nonzero entries that are independently and uniformly distributed in the interval $[-1, 1]$. The regularization parameters β and γ are set to $0.0005n$. In view of Theorem 4.2.2, Theorem 4.4.1 and Theorem 5.3.3, T-BD, Acc-BD and A-SP-HPE generate an easily computable SP-residual $((\tilde{r}_k^x, \tilde{r}_k^y), \varepsilon_k^x + \varepsilon_k^y)$ at each iteration. We have also implemented a version of Korp (see for example [35]) that generates the above easily computable SP-residuals. The above four methods are then terminated whenever

$$\max\left\{\frac{\|(\tilde{r}_k^x, \tilde{r}_k^y)\|}{\max\{1, \|\tilde{x}_k\|, \|\tilde{y}_k\|\}}, \varepsilon_k^x + \varepsilon_k^y\right\} < \epsilon. \tag{164}$$

Table 3 reports the CPU time and the number of singular value decomposition (in order to evaluate the resolvent of ∂g_2) for the above four methods. Due to space limitation, Table 2 does not specify the number of iterations performed by each method. We note however that the number of iterations performed by T-BD and Korp can be obtained by dividing the corresponding number of SVD computations by 1 and 2, respectively. Also, the number of outer (HPE) iterations performed by Acc-BD is equal to the number of SVD computations due to the fact that $L_{yy} = 0$ for the saddle-point considered in this subsection and the fact that the safeguard used in our

implementation ensures that the proximal subproblem in the y-variable is solved by means of a single resolvent evaluation of ∂h_2^*.

Table 3 shows that Korp and A-SP-HPE was the slowest among the four methods on 8 out of 9 instances. The computational results also show that Acc-BD was the fastest on this collection, and it performed especially well when the computational cost of computing a SVD is much larger than that of a matrix-vector multiplication.

Table 3: Computational results for the methods Acc-BD, A-SP-HPE, T-BD and Korp on the regularized least-square problems (163) with different problem sizes. The four methods are terminated using criterion (164) with $\epsilon = 10^{-3}$. CPU time in seconds and the number of singular value decomposition are reported for each method.

Problem				Acc-BD		T-BD		A-SP-HPE		Korp	
m	k	n	$\frac{L_{xx}}{L_{xy}}$	time	#svd	time	#svd	time	#svd	time	#svd
100	100	100	2.40	**0.31**	35	0.35	72	1.40	442	1.23	276
100	200	200	4.14	**1.70**	36	1.88	93	5.42	340	5.96	356
100	500	500	4.92	**62.71**	99	191.43	465	225.18	594	265.85	752
200	100	100	3.79	**0.61**	50	0.87	165	3.85	706	3.62	636
200	200	200	5.44	**1.60**	39	2.65	106	5.81	258	8.64	398
200	500	500	7.14	**44.63**	100	175.11	682	189.01	872	376.03	1350
500	100	100	5.70	**0.28**	26	0.61	91	1.59	284	2.48	350
500	200	200	6.61	**1.72**	38	3.23	108	8.67	362	10.85	414
500	500	500	9.49	**57.06**	126	169.56	681	211.79	1300	577.10	2558

6.4 Real-world Applications

In the section, we evaluate the performance of the two proposed methods Acc-BD, A-SP-HPE for solving the real-world applications introduced in Section 2.3.

6.4.1 Sparse PCA

This subsection compares the performance of Acc-BD and A-SP-HPE to Korpelevich's extragradient method (Korp) for solving sparse PCA problem (31) on five real-world data sets which were collected and studied in [26]. Table 4 reports the name of each data set and their corresponding dimension of the matrix A in problem (31). In our numerical experiment, the given covariance matrix A is scaled as that its largest eigenvalue is 1 and the parameter ρ in (31)is set as 0.001.

The three methods compared in this subsection are terminated whenever the duality gap (22) at the candidate solution $(\tilde{x}, \tilde{y})$ is less than a given tolerance ϵ. Both the iterate sequence $\{(\tilde{x}_k, \tilde{y}_k)\}$ and the ergodic sequence $\{(\tilde{x}^a_k, \tilde{y}^a_k)\}$ are used to check the duality gap (22) for the three methods considered in this section. Table 4 reports the CPU time and the number of (outer) iterations for each method.

Table 4: Computational results for the methods Acc-BD, A-SP-HPE and Korp on sparse PCA problems with different problem sizes. The three methods are terminated whenever the duality gap (22) is less than $\epsilon = 10^{-2}$. CPU time and the number of (outer) iterations are reported for each method.

Data set	problem size	Acc-BD	A-SP-HPE	Korp
	#variables	time / iter.	time / iter.	time / iter.
Arabidopsis thaliana	834	**527.07 / 1490**	1407.3 / 2870	1173.8 / 1860
Estrogen receptor	692	**163.30/ 715**	426.57 / 1375	283.07 /700
Leukemia	1255	**1033.9/990**	2728.5/1910	2440.7 / 1290
Lymph node status	587	**87.6 / 550**	226.97 / 1055	152.51 / 535
Hereditary breast cancer	1869	**6242.9 / 1925**	16813.2/3750	21407.3 /3485

6.4.2 Sparse inverse covariance estimation

This subsection compares the performance of Acc-BD and A-SP-HPE to Korpelevich's extragradient method (Korp) for solving a collection of instances of SICE problem (33) on the same five real-world data sets used in Subsection 6.4.1. In our numerical experiment, the given covariance matrix A is scaled as that its largest eigenvalue is 1 and the matrix Λ in (33) is set as 0.005 for off-diagonal elements and 0 for diagonal elements.

The three methods compared in this subsection are terminated whenever the duality gap (22) at the candidate solution $(\tilde{x}, \tilde{y})$ is less than a given tolerance ϵ. Both the iterate sequence $\{(\tilde{x}_k, \tilde{y}_k)\}$ and the ergodic sequence $\{(\tilde{x}^a_k, \tilde{y}^a_k)\}$ are used to check the duality gap (22) for the three methods considered in this section. Table 5 reports the CPU time and the number of (outer) iterations for each method.

Table 5: Computational results for the methods Acc-BD, A-SP-HPE and Korp on SICE problems with different problem sizes. The three methods are terminated whenever the duality gap (22) is less than $\epsilon = 10^{-1}$. CPU time and the number of (outer) iterations are reported for each method.

Data set	problem size	Acc-BD	A-SP-HPE	Korp
	#variables	time / iter.	time / iter.	time / iter.
Arabidopsis thaliana	834	**89.22/235**	255.98/ 335	716.65 / 1110
Estrogen receptor	692	**50.31/ 205**	154.46/ 305	440.64 / 1015
Leukemia	1255	**652.80/595**	1998.2/890	4741.3/ 1985
Lymph node status	587	**25.77/ 150**	78.29 /220	233.35/ 795
Hereditary breast cancer	1869	**1578.9 / 1510**	5030.8/ 2325	4263.9/3115

6.4.3 Truncated collaborative filtering for recommender system

This subsection compares the performance of Acc-BD and A-SP-HPE to Nesterov's method and Korpelevich's extragradient method (Korp) for solving four instances of the truncated collaborative filtering problem (36) where the matrix R are provided by three synthetic data sets and the real-world movielens data set [1]. The dimensions of matrix R in these four instances are $200 \times 200, 500 \times 500, 1000 \times 1000$ and 2000×3000 respectively. In the synthetic data sets, R was generated as the sum of a low-rank matrix (rank=50) and random noise, which are then truncated to the range [1,5] and S is generated as binary matrix with 5% nonzero elements. The parameters in problem (36) are set as $\mu = 0.05, l = 1$ and $u = 5$.

Instead of terminating the four methods using termination criteria, we fix the number of outer iterations as 100 and plot the objective function v.s. time to compare the four methods. As shown in Figure 1, while Ac-BD and Korp are the fastest two methods, Acc-BD is slighted better than Korp on this set of problems and A-SP-HPE is substantially better than Nesterov' method.

[1] http://movielens.org

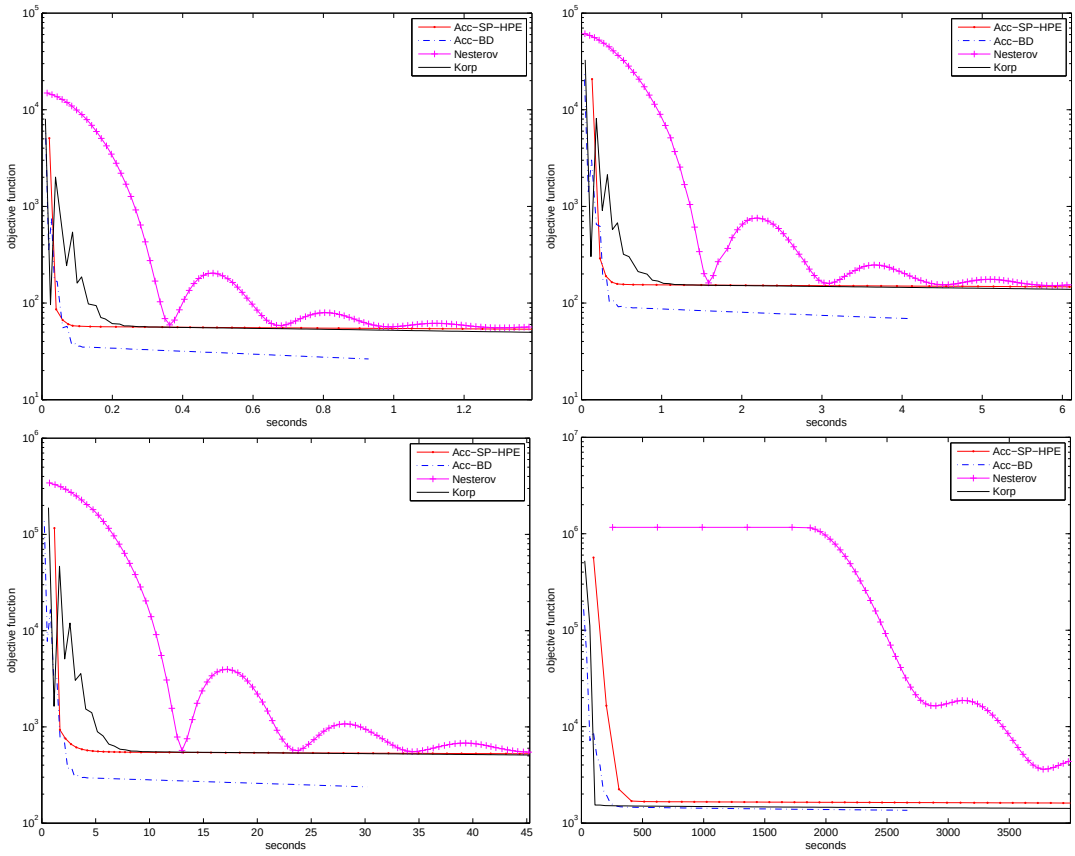

Figure 1: Computational results for the methods Acc-BD, A-SP-HPE, Nesterov's method and Korp for solving truncated collaborative filtering problem (36). The plots report objective function value v.s.time comparison on four data sets of size $200 \times 200, 500 \times 500, 1000 \times 1000$ and 2000×3000 respectively.

6.4.4 MR image recovering

This subsection compares the performance of Acc-BD and A-SP-HPE to Nesterov's method and Korpelevich's extragradient method (Korp) for solving the MR image recovering problem (37) on the four real-world images. For all four cases, we create the noisy observation using a partial Fourier convolution and a masking operator to hide 40% pixels. The additive white Gaussian noise has standard deviation $\sigma = 0.025$. The reconstruction operator W uses separable bidimensional Daubechies wavelets with two vanishing moments which generate a dictionary with redundancy $l = 13$. The parameters in problem (36) are set as $\mu = 0.0005$ and $\nu = 0.005$.

Instead of terminating the four methods using termination criteria, we fix the

number of outer iterations as 100 and plot the objective function v.s. time to compare the four methods. As shown in Figure 2 – 5, while Ac-BD and Korp are the fastest two methods, Acc-BD is slighted better than Korp on this set of problems and A-SP-HPE is substantially better than Nesterov' method.

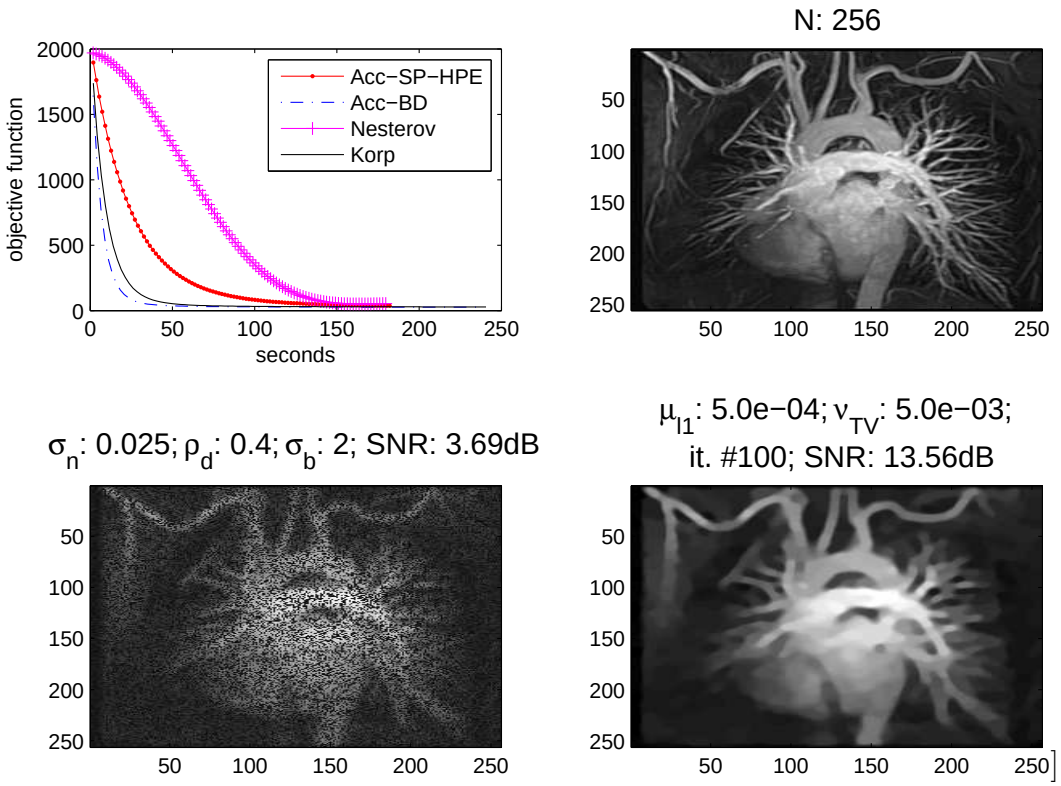

Figure 2: Recovering chest MR image: [Upper left] Computational results for the methods Acc-BD, A-SP-HPE, Nesterov's method and Korp for solving MR image recovering problem (37); [upper right] original Image; [bottom left] observed image; [bottom right] image recovered by Acc-BD.

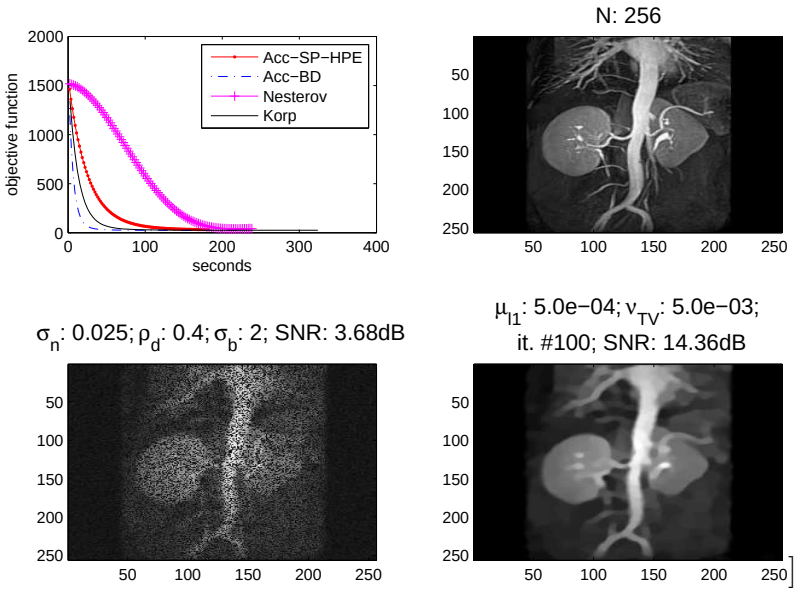

Figure 3: Recovering Renal Arteries MR image: [Upper left] Computational results for the methods Acc-BD, A-SP-HPE, Nesterov's method and Korp for solving MR image recovering problem (37); [upper right] original Image; [bottom left] observed image; [bottom right] image recovered by Acc-BD.

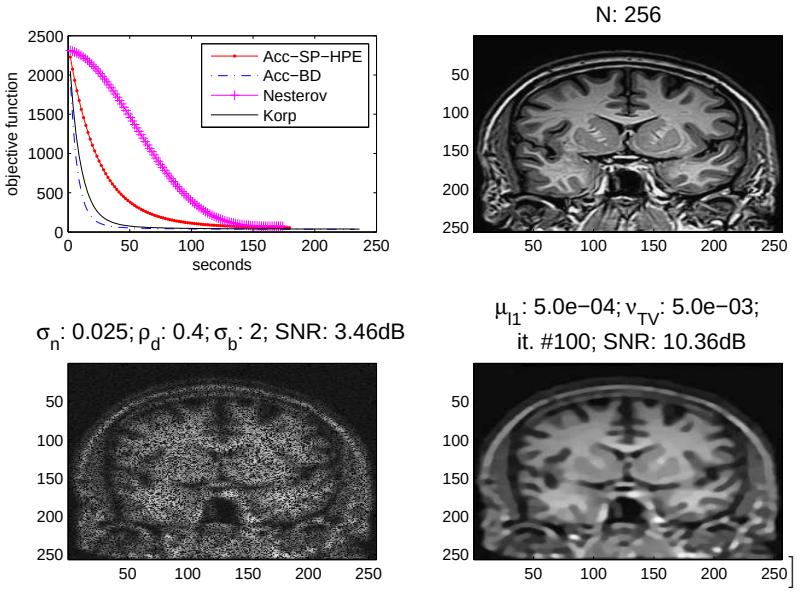

Figure 4: Recovering coronal brain MR image: [Upper left] Computational results for the methods Acc-BD, A-SP-HPE, Nesterov's method and Korp for solving MR image recovering problem (37); [upper right] original Image; [bottom left] observed image; [bottom right] image recovered by Acc-BD.

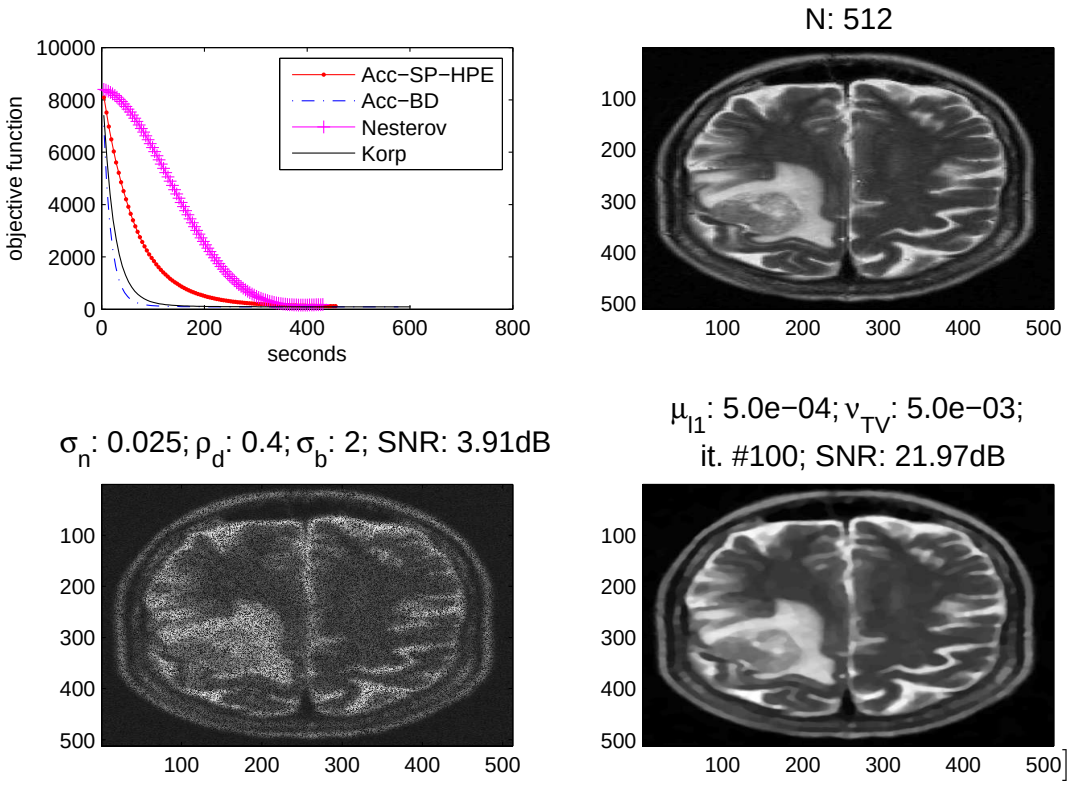

Figure 5: Recovering brain MR image: [Upper left] Computational results for the methods Acc-BD, A-SP-HPE, Nesterov's method and Korp for solving MR image recovering problem (37); [upper right] original Image; [bottom left] observed image; [bottom right] image recovered by Acc-BD.

CHAPTER VII

CONCLUDING REMARKS

7.1 Summary of contributions

In this dissertation, two new methods are introduced for solving composite saddle-point problems. The first method, Acc-BD, is a special instance of the BD-HPE framework for solving CSP problem (5). It exploits the fact that the two prox sub-inclusions are equivalent to composite convex programs. By using a Nesterov-type accelerated method to approximately solve them. It is shown that the new method outperforms previous BD-HPE methods both theoretically and computationally in situations where $\max\{L_{xx}, L_{yy}\} >> L_{xy}$. Moreover, The experiment results on seven problem sets have shown that the new method significantly outperforms several state-of-the-art algorithms.

The second new algorithm, A-SP-HPE, is proposed for solving a special class of CSP problems (6). This method is a special instance of the hybrid proximal extragradient (HPE) framework in which a Nesterov's accelerated variant is used to approximately solve the prox subproblems. One of the advantages of the this method is that it works for any constant choice of proximal stepsize. Moreover, a suitable choice of the latter stepsize yields a method with the best known (accelerated inner) iteration complexity for the aforementioned class of saddle-point problems. In contrast to the smoothing technique of Nesterov, this new accelerated method does not assume that feasible set is bounded due to its proximal point nature.

Last but not least, this dissertation demonstrates a few examples of real-world applications in the area of machine learning and image processing which can be formulated as composite saddle-point problems and then can take advantage of the

recent development in the area of numerical optimization for saddle-point problems.

7.2 Future work and challenges

First, it is worth noting that both the mirror-prox method by Nemirovskii [39] and Nesterov's smooth approximation scheme [40] can be implemented using the entropy distance-generating function and with $\mathcal{X}$ and $\mathcal{Y}$ endowed with the L_1-norm. In the future, we plan to design variants of Acc-BD and Acc-SP-HPE that can take advantage of entropy distance-generating function and compare it with the corresponding variants of mirror-prox method and Nesterov's method.

Second, it is still an open problem that if HPE and BD-HPE framework can be extended for for solving stochastic saddle-point problems. The challenge of this direction is due to that the formulation of both frameworks uses error criteria (3) which must be satisfied at every iteration. Since the stochastic setting of composite saddle-point problem is extremely important for real-world applications, especially large-scale machine learning problems, we would like to continue to explore this direction in the future.

REFERENCES

[1] Auslender, A. and Teboulle, M., *Asymptotic cones and functions in optimization and variational inequalities*. Springer Monographs in Mathematics, New York: Springer-Verlag, 2003.

[2] Beck, A. and Teboulle, M., "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[3] Beck, A. and Teboulle, M., "Smoothing and first order methods: a unified framework," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 557–580, 2012.

[4] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers, 2011.

[5] Burachik, R. S., Iusem, A. N., and Svaiter, B. F., "Enlargement of monotone operators with applications to variational inequalities," *Set-Valued Anal.*, vol. 5, no. 2, pp. 159–180, 1997.

[6] Chambolle, A. and Pock, T., "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[7] Chen, Y., Lan, G., and Ouyang, Y., "Optimal primal-dual methods for a class of saddle point problems," *arXiv preprint arXiv:1309.5548*, 2013.

[8] Combettes, P. L. and Pesquet, J.-C., "Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators," *Set-Valued and variational analysis*, vol. 20, no. 2, pp. 307–330, 2012.

[9] Condat, L., "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013.

[10] d Aspremont, A., El Ghaoui, L., Jordan, M., and Lanckriet, G., "A direct formulation for sparse PCA using semidefinite programming," *SIAM review*, vol. 49, no. 3, p. 434, 2007.

[11] d'Aspremont, A., Bach, F., and Ghaoui, L., "Optimal solutions for sparse principal component analysis," *The Journal of Machine Learning Research*, vol. 9, pp. 1269–1294, 2008.

[12] Eckstein, J. and Svaiter, B. F., "A family of projective splitting methods for the sum of two maximal monotone operators," *Math. Program.*, vol. 111, no. 1-2, Ser. B, pp. 173–199, 2008. Published as IMPA Tech. Rep. A 238/2003 in 2003.

[13] Eckstein, J. and Svaiter, B. F., "General projective splitting methods for sums of maximal monotone operators," *SIAM J. Control Optim.*, vol. 48, no. 2, pp. 787–811, 2009.

[14] Friedman, J., Hastie, T., and Tibshirani, R., "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[15] He, N., Juditsky, A., and Nemirovski, A., "Mirror prox algorithm for multi-term composite minimization and alternating directions," *arXiv preprint arXiv:1311.1098*, 2013.

[16] He, Y. and Monteiro, R. D., "Accelerating block-decomposition first-order methods for solving composite nash equilibrium and saddle-point problems," *Submitted to SIAM Journal on Optimization*, 2013.

[17] He, Y. and Monteiro, R. D., "An accelerated hpe-type algorithm for a class of composite convex-concave saddle-point problems," *Submitted to SIAM Journal on Optimization*, 2014.

[18] He, Y., Monteiro, R. D., and Park, H., "An algorithm for sparse pca based on a new sparsity control criterion," in *Proceedings of the SIAM International Conference on Data Mining, Mesa, AZ*, SIAM, 2011.

[19] He, Y., Qi, Y., Kavukcuoglu, K., and Park, H., "Learning the dependency structure of latent factors," in *Advances in Neural Information Processing Systems*, pp. 2366–2374, 2012.

[20] Hsieh, C., Sustik, M., Ravikumar, P., and Dhillon, I., "Sparse inverse covariance matrix estimation using quadratic approximation," *Advances in Neural Information Processing Systems (NIPS)*, vol. 24, 2011.

[21] Jacob, L., Obozinski, G., and Vert, J.-P., "Group lasso with overlap and graph lasso," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 433–440, ACM, 2009.

[22] JournŽe, M., Nesterov, Y., Richtarik, P., and Sepulchre, R., "Generalized power method for sparse principal component analysis," *CORE Discussion Papers*, 2008.

[23] Kannan, R., Ishteva, M., and Park, H., "Bounded matrix low rank approximation," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pp. 319–328, IEEE, 2012.

[24] Kolda, T. G. and Bader, B. W., "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[25] Korpelevič, G. M., "An extragradient method for finding saddle points and for other problems," *Èkonom. i Mat. Metody*, vol. 12, no. 4, pp. 747–756, 1976.

[26] Li, L. and Toh, K.-C., "An inexact interior point method for l 1-regularized sparse covariance selection," *Mathematical Programming Computation*, vol. 2, no. 3-4, pp. 291–315, 2010.

[27] Lu, Z. and Zhang, Y., "An Augmented Lagrangian Approach for Sparse Principal Component Analysis," *Arxiv preprint arXiv:0907.2079*, 2009.

[28] Lu, Z., "Smooth optimization approach for sparse covariance selection," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1807–1827, 2009.

[29] Ma, S., Yin, W., Zhang, Y., and Chakraborty, A., "An efficient algorithm for compressed mr imaging using total variation and wavelets," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[30] Mackey, L., "Deflation methods for sparse pca," *Advances in Neural Information Processing Systems*, vol. 21, pp. 1017–1024, 2009.

[31] Mairal, J., Jenatton, R., Obozinski, G., and Bach, F., "Convex and network flow optimization for structured sparsity," *The Journal of Machine Learning Research*, vol. 12, pp. 2681–2720, 2011.

[32] Mazumder, R., Hastie, T., and Tibshirani, R., "Spectral regularization algorithms for learning large incomplete matrices," *The Journal of Machine Learning Research*, vol. 11, pp. 2287–2322, 2010.

[33] Moghaddam, B., Weiss, Y., and Avidan, S., "Spectral bounds for sparse PCA: Exact and greedy algorithms," *Advances in Neural Information Processing Systems*, vol. 18, p. 915, 2006.

[34] Monteiro, R. D. C., Ortiz, C., and Svaiter, B. F., "Implementation of a block-decomposition algorithm for solving large-scale conic semidefinite programming problems," *Optimization-online preprint*, vol. 3032, pp. 1–32, 2011.

[35] Monteiro, R. D. C. and Svaiter, B. F., "On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 2755–2787, 2010.

[36] Monteiro, R. D. C. and Svaiter, B. F., "Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 475–507, 2013.

[37] Monteiro, R. D. C. and Svaiter, B., "Complexity of variants of Tseng's modified F-B splitting and korpelevich's methods for hemivariational inequalities with applications to saddle-point and convex optimization problems," *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1688–1720, 2011.

[38] Monteiro, R. D., Ortiz, C., and Svaiter, B. F., "A first-order block-decomposition method for solving two-easy-block structured semidefinite programs," *to appear in Mathematical Programming Computation*, 2013.

[39] Nemirovski, A., "Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 229–251, 2004.

[40] Nesterov, Y., "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[41] Nesterov, Y., "Dual extrapolation and its applications to solving variational inequalities and related problems," *Mathematical Programming*, vol. 109, no. 2-3, pp. 319–344, 2007.

[42] Nesterov, Y., "Gradient methods for minimizing composite functions," *Mathematical Programming*, pp. 1–37, 2012.

[43] Ouorou, A., "Epsilon-proximal decomposition method," *Math. Program.*, vol. 99, no. 1, Ser. A, pp. 89–108, 2004.

[44] Raguet, H., Fadili, J., and Peyré, G., "Generalized forward-backward splitting," *arXiv preprint arXiv:1108.4404*, 2011.

[45] Rockafellar, R. T., "On the maximal monotonicity of subdifferential mappings," *Pacific J. Math.*, vol. 33, pp. 209–216, 1970.

[46] Rockafellar, R. T., "Monotone operators and the proximal point algorithm," *SIAM J. Control Optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[47] Rockafellar, R. T. and Wets, R. J.-B., *Variational Analysis*, vol. 317. Springer, 1998.

[48] Rudin, L. I., Osher, S., and Fatemi, E., "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[49] Scheinberg, K., Ma, S., and Goldfarb, D., "Sparse inverse covariance selection via alternating linearization methods," *Advances in Neural Information Processing Systems (NIPS)*, vol. 23, 2010.

[50] Shen, H. and Huang, J., "Sparse principal component analysis via regularized low rank matrix approximation," *Journal of multivariate analysis*, vol. 99, no. 6, pp. 1015–1034, 2008.

[51] Solodov, M. V., "A class of decomposition methods for convex optimization and monotone variational inclusions via the hybrid inexact proximal point framework," *Optim. Methods Softw.*, vol. 19, no. 5, pp. 557–575, 2004.

[52] Solodov, M. V. and Svaiter, B. F., "A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator," *Set-Valued Anal.*, vol. 7, no. 4, pp. 323–345, 1999.

[53] Solodov, M. V. and Svaiter, B. F., "A hybrid projection-proximal point algorithm," *J. Convex Anal.*, vol. 6, no. 1, pp. 59–70, 1999.

[54] Solodov, M. V. and Svaiter, B. F., "An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions," *Math. Oper. Res.*, vol. 25, no. 2, pp. 214–230, 2000.

[55] Solodov, M. V. and Svaiter, B. F., "A unified framework for some inexact proximal point algorithms," *Numer. Funct. Anal. Optim.*, vol. 22, no. 7-8, pp. 1013–1035, 2001.

[56] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K., "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[57] Tomioka, R., Suzuki, T., Hayashi, K., and Kashima, H., "Statistical performance of convex tensor decomposition," in *Advances in Neural Information Processing Systems*, pp. 972–980, 2011.

[58] Tseng, P., "A modified forward-backward splitting method for maximal monotone mappings," *SIAM J. Control Optim.*, vol. 38, no. 2, pp. 431–446 (electronic), 2000.

[59] Tseng, P., "A modified forward-backward splitting method for maximal monotone mappings," *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 431–446, 2000.

[60] Tseng, P., "On accelerated proximal gradient methods for convex-concave optimization," *submitted to Journal on Optimization*, 2008.

[61] Vũ, B. C., "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Advances in Computational Mathematics*, vol. 38, no. 3, pp. 667–681, 2013.

[62] Zou, H., Hastie, T., and Tibshirani, R., "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

VITA

Yunlong He was born in Xiangfan, Hubei Province, China in 1989. He finished his B.S. in Mathematics at the age of 19, graduating from the Special Class for Gifted Young in USTC. He received his M.S. degree in Statistics in 2013 from Georgia Tech. He has been studying machine learning and numerical algorithms since 2008 and has worked with prominent scholars in the areas of numerical algorithms, optimization, machine learning, statistics and computer vision. He works as a research scientist in Bay Area after receiving his PhD degree in Computational Science and Engineering in 2014.